

THE NEW COLOR COMPUTER

by Bruce Warner

By the time this article hits the street, I'll have seen the new Color Computer. On July 30th, 1986, a press conference was hosted in New York City. The conference introduced four new MS-DOS computers and the new Tandy Color Computer.

This machine supports advanced graphics, up to 640 x 192 pixels in FOUR colors. The majority of existing hardware is compatible, including Deluxe RS-232 Paks and disk controllers, but much of the hardware may be undesired with the improved features of this machine. Incompatibilities will come from hardware and software that has violated specific conventions of programming standards both under

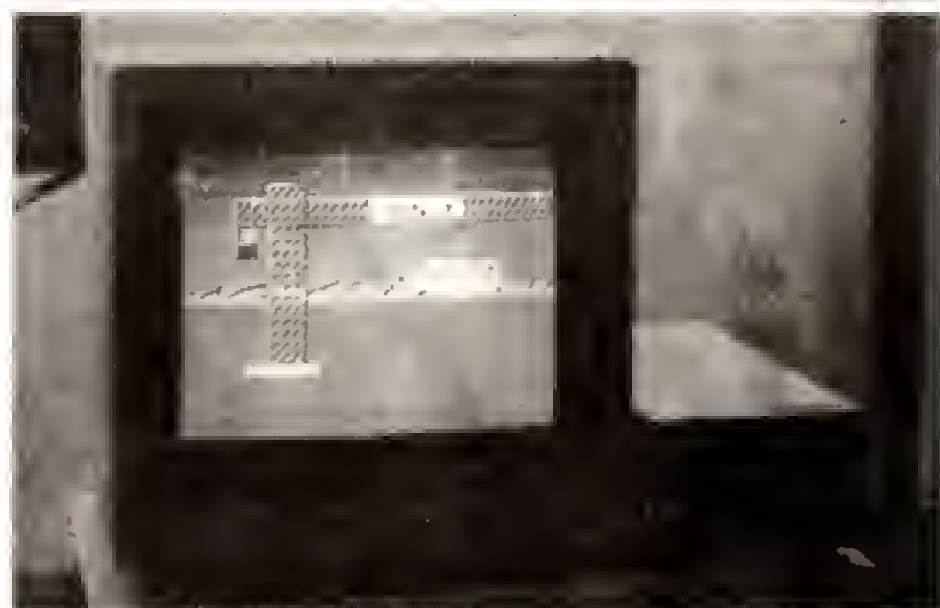
OS-9 and RS-DOS. Most of the programs are the ones that run in the "96K" mode, or those that make direct calls to hardware devices rather than using device descriptors and drivers. CoCo Max is among the programs that will not run, due to the 96K addressing used by CoCo Max. Most of the Tandy hardware will work without modification, since they have followed their own programming rules. An exception to the Tandy rule is the Multi-Pak interface under OS-9 Level II. A \$6 modification is required.



From the outside, the new Tandy Color Computer looks a lot like its forerunner, the Color Computer 2, but the insides make all the difference in the world.

The Software Compatibility Story

If you're using Telewriter under RS-DOS, you'll be surprised that it is one of the few programs that won't have compatibility problems. Telewriter-64 works because of its implementation of the 64K CoCo. Although the original version used undocumented calls to the ROM, which resulted in incompatibility of the software, the current version works because they didn't try to make Telewriter-64 work using 96K. This is due to the addressing mode of the GIMI chip. This is the chip that gives you all the power to



There's blue sky (four shades), a red crane, a black, white and red CoCo 3, green hills, orange foreground and black text. Altogether, 16 colors are available with higher resolution than the old CoCo and more programming memory than the old graphics modes.

address the extra memory. To make a "96K" program run, the GIMI chip must be set to 16K internal and 16K external ROM before performing any banking between the RAM/ROM and all RAM mode using the usual SAM addressing. The easiest way around this is to go to OS-9 and let the new Level II OS-9 system handle it all for you. This will also insure that your software runs first time out on any new Tandy OS-9 machine. RS-DOS also causes some problems with addressing memory that is reserved for the GIMI and SAM chips.

Tandy has listened to us, so you'll see more than you may have expected in a Color Computer, with memory starting at 128K and working its way up to 512K (in one step) and a variety of features. You'll even see graphics that rival the outhouse and game companies. We'll be able

to see this machine yourself in your local Radio Shack in a couple of weeks.

So what's so new and so great about a 6809 computer? Well, you may decide that you don't even need a 68xxx machine after seeing this one. Mostly because of the things that have made the OS-9 community so great to begin with, but with the added features of this machine it's a real bargain.

Besides the memory, you'll be glad to know that BASIC is still BASIC for those that want to stay that way. What makes BASIC better is that Microware (that's right, the OS-9 people) were contracted to do the patch for Enhanced Color BASIC. This includes 21 new commands aimed at making the best use of the hardware in this machine. This includes the new 80-character

Continued on Page 14

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 18
PROSPECT, KY

86/09/13

OS-9 Users Group
Attn: Membership
9743 University Ave., Ste. 330
Des Moines, IA 50322

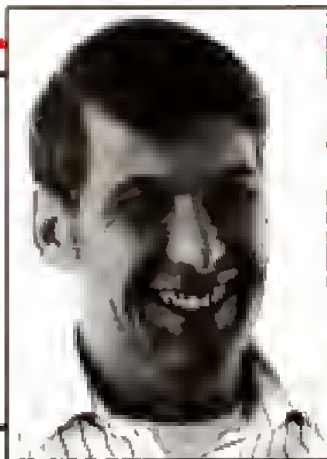
Address Correction Requested

OS-9 AT NCC

(Photos and story — Pages 10 and 11)

Richard Don, owner of Gimix, Inc., shows off the Gimix that fits in a PC case. The use of the PC power supply and case offers the user the option of an exceptionally low-priced casing for their single-board computer.





PUTTING IT TOGETHER

The building of this month's MOTD

by Bruce N. Warner, Editor



September, 1986

Vol. 2, No. 7

MOTD is published monthly as the official newsletter of the National OS-9 Users Group, 9743 University Ave., Suite 330, Des Moines, IA 50322

The building of this month's MOTD may have been a little harder than usual. We'll start with the stupid complaints by people that have proclaimed themselves as OS-9 experts and go on to the daily routine of helping a friend whose son wants to join the Navy.

What in the world do these things have in common? Simple. They are two of the many things that occupy my world, from the few petty folks in the world of OS-9 (that take up most of your time) to the people that really have a problem and want a little shove in the right direction.

Puckett Jumps on the Soap Box

Dale Puckett has an excellent article this month on OS-9 critics. They remind me of Ralph Nader. Though Mr. Nader has spent a lot of time making us aware of how unsafe our cars and other items are, I sure am glad that I don't have to wait for the car he designs. I'm also glad I'm allowed to take some responsibility and not have to pay the extra \$100,000 it would take to put all the extra features Mr. Nader would want as standard equipment on our cars.

Dale hits the nail on the head when he tells us that "Those who can, do. Those who can't, criticize." If you're a critic, try being a doer. It doesn't take that much more energy, and it sure gets us into a healthier mood.

A Week in Vegas

Trying to cover everything in the world of OS-9 isn't easy. I just spent every penny I made editing the MOTD this year by going to one show. Not all of the shows are paid for by the UG, and the editor's post is the only paid position. The pay is less than adequate for what it takes, and I could make more spending the same amount of time writing for our local newspaper. The point is that I could be doing something a lot more profitable and much easier with my time. Like so many of the UG officers, I believe that OS-9 is worth the effort.

In case you're wondering why I refer to going to Las Vegas as an expense, I hit on the fact that I don't gamble and I don't drink, so the shows were nice, but the location isn't my idea of a good time. I really would have better enjoyed a week at a riding academy up north in June. The desert heat isn't my idea of great weather.

Coming up This Fall

If you're wondering about RAINBOWfest, you'll be pleased to know that we're all very proud to announce that the publishers of the RAINBOW are planning an expanded coverage of OS-9 seminars. Some of these will be because of the new Color Computer (see the article in this issue), others are because of an increased interest in OS-9.

I'm very proud to announce that I'll be giving a seminar on "OS-9 From a Users Standpoint." This talk will be dedicated to finding out what OS-9 means to those people who only want to use the software available on their computer. I, for one, rarely do any programming under OS-9. Instead, I purchase programs that are already written under our favorite operating system and learn how to use them and make the system work better for us all. I also get software and hardware from various companies and write the documentation for making it run under OS-9. This development of hardware and software documentation is one of my largest uses of OS-9, and it's greatest visibility is in the Color Computer OS-9 community.

We Don't Always Get to Talk

There are some people that think the UG officers talk on a daily basis. There are times that we don't even get to talk on a weekly basis. I haven't spoken to Brian Lantz but once in the last month. I do get to talk with Dale Puckett at least once a week, but Dale lives and works in the local area, and phone calls are cheap for us.

What's all this leading up to? Well, don't be surprised if you get different stories from different officers in the same day. We may get our heads together before the next time you give us a call, but at any given moment, we are just as likely as you to be operating on different frequencies. Add to that, there is always more than one way to look at and solve a problem. At one RAINBOWfest, Dale Puckett and I were answering several questions for people. Dale gave one set of answers and I gave another. While we were both right, there were times that Dale's answers were better and times they were worse than mine. They were all right, mind you, only looked at from a different person with a different point of view.

I keep remembering this incident when someone asks me what is the

best word processor for their OS-9 computer. I tell them I can't answer that, but I use DynaStar.

The next question is always, "Then why do you use DynaStar if it's not the best?" Simple. It is the best — for me! What's best for you could be StyloGraph or Last Word or DeskMate. I don't know what you're going to do with it or how you're going to make it work for you. I only know what's right for me. So, is my answer wrong? No. My answer is right under the circumstances I've chosen to consider to answer it. I use DynaStar because it's comfortable and it handles about 98 percent of my word processing needs. What you select is your option. What I select is mine. With the new Color Computer, we may find some really great word processors hitting the OS-9 community.

The point is that we can only answer your questions from our own vantage point.

Never Become an Expert

I have a friend who says an "EX" is a has-been and a "SPERT" is a drip under pressure, so what's that make an EX-SPERT? There's a much better reason for not becoming an expert. There is always someone out there that wants to rip your head off. As soon as your name becomes known, you're either a target or a source of information. It's kind of like being the fastest gun in the West. There's always someone that wants to shoot you out of the fastest seat.

I know of people that quote Steve Bjork, Dale Puckett and Brian Lantz for all sorts of things. One told me that Dale Puckett had been invited to the Tandy press conference in New York. I was glad to hear it, but a bit surprised. Dale and I had just spoken the day before and he had no plans or invitation to attend the conference. I, on the other hand, was planning a one-day trip to New York for the event.

The next day, Dale showed up at the monthly Saturday meeting of the Northern Virginia Color Computer Club, much to my surprise. What happened next was a little embarrassing for the member that told me Dale was going. Dale hadn't spoken with him or the people at Tandy. There was an assumption made because of Dale's reputation in the field and a desire to be asso-

Continued on Page 20

Editor

Bruce Warner
OS-9 Users Group
ATTN: Editor
9743 University Ave., Suite 330
Des Moines, IA 50322

President, National OS-9 Users Group
Brian Lantz

Vice President
Bill Turner

Treasurer
Steve Odneal

Secretary
Dave Gibson

Director-at-Large
Dale Puckett

Software Librarian
Dave Kaleita

Finance Committee
George Dorner — Chairman

Membership Committee
Joe Dubuc — Chairman

Public Relations Committee
Bob Rosen — Chairman

Software Exchange Committee
Carl Kreider — Chairman

Software Reviews Committee
Marcus Johnson — Chairman

Subscriptions to the MOTD are free to members of the OS-9 Users Group. Non-members may subscribe by writing to the chairman of the membership committee and paying an annual fee of \$20. Send to: OS-9 Users Group, ATTN: Membership, 9743 University Ave., Suite 330, Des Moines, IA 50322.

MOTD is produced and distributed by Falsolt, Inc., publishers of THE RAINBOW, the official magazine of the National OS-9 Users Group.

OS-9 is a registered trademark of Microware. There is no connection between Microware and The National OS-9 Users Group, nor is there any connection between Falsolt and The National OS-9 Users Group.

MOTD Advertising
c/o Bruce N. Warner
14503 Fullerton Road
Dale City, VA 22193-2034
Phone (703) 670-4962
(Include check payable to the OS-9 Users Group)



NEW LOW COST 68020 BASED MULTI-USER SUPER-MICRO

The **QT 20x™** is an extremely cost effective multi-user, multi-tasking micro that will support 2 to 30 users. RAM is expandable from 512K to 14.5 Megabytes; ROM is expandable to 256K. A number of useful software packages are included in its low cost. The **QT 20x™** can be purchased as a complete system with floppy and hard drives; in a kit form, or the motherboard can be purchased separately. Other hardware configurations can be customized to meet your needs. Listed below are the principal hardware and software features:

HARDWARE Features:

- 32 Bit MC68020 Processor - 12.5 MHz
- 512 Kbytes of 32-bit wide, high speed RAM
- 2 RS-232 Serial I/O Ports
- 1 Parallel I/O Port
- Built-In SCSI/SASI Interface With DMA
- Floppy controller with DMA
- 32 Bit Expansion Buss
- Real Time Clock w/Battery Backup
- DIP Switches to Select Boot Up Baud Rates and Boot Device

HARDWARE Options:

- 2 Megabyte RAM / 4 Serial Port Expansion, (a maximum of 7 boards may be used)
- MC68881 Floating-point Coprocessor
- Memory Protection **
- Graphics Board **
- 4 Serial Port Expansion Board **

** (Available 4th qtr)

SOFTWARE included:

- OS9/68K - Operating System
- Basic09 - Programming Language
- Stylo - Word Processing
- Mail Merge
- Spelling Checker
- Dynacalc - Spreadsheet
- fbu - Hard Disk to Floppy Backup ***
- QCom - Communications with XMODEM and text file transfer ***

Other Software Available:

- 'C'
- Pascal
- PC Utility (Read/Write/Format IBM Disks)
- CRASMB Cross Assembler
- Sculptor Plus - 4th Generation P.L. (Data Base Management)
- Fortran 77 (Available 4th qtr)

Note: *** QCom & fbu are included FREE when a complete system is purchased.

LIMITED TIME SPECIAL

Complete system with 20
Meg Drive only **\$4495!**

2.5 Meg RAM, 6 Serial Ports 1 Parallel port, RTC.

1976  FRANK HOGG
LABORATORY 1986
770 James St., Syracuse, NY 13203 Telex 646740
315/474-7856

OS-9 U G SOFTWARE COMMENTARY

by Bert Schneider

Wow! 39,168 sectors — I never thought I'd see that many sectors on my Color Computer. I finally got the hard drive up and running on my system thanks to people like Ron Schmidt of L.R. Tech! The only problem I have now is that I have been unable to get OS-9 Version 2.0 to boot up on my system using the hard drive (I have all the upgrades for 2.0). Anyhow, I can at least boot up 1.1 without any floppy drives.

NEW UPDATES! Some of the User Group Library Disks have been updated to correct bugs or add more efficient code, or just to add some new routines.

Dave Kaleita has improved his documentation generation program DOCGEN2 (a program used by software authors who want to contribute to the OS-9 UG Library). Dave has included on Disk #0 a new version called DOCGEB3. He states that it cannot be packed on a CoCo, but I did it on mine without problems (I am also using Wordpak which frees up a great deal of memory). He also includes a new utility called LOWUP that converts all text to uppercase only. His DOCGEN3 program requires all uppercase for data input.

Disks #6 and #7 have been fixed. The adventure program had a few bugs that would really irritate an avid adventurer. After spending hours down in Colossal Cave, you might suddenly find your adventure curtailed by a FATAL ERROR \$38 or other such messages. Those have been corrected on both the source code disk and the object code disk. Other disks in the library have been updated too. Since I have not covered those yet, I will just treat the new updated versions as the standard baseline disk for commentary purposes.

Disk #10, "Math and Electronics," fits right up my alley since I am an electrical engineer. All of the code on this disk is written in Basic09 and was put together by George Dorner (another OS-9 UG superman). I only wish I had access to some of these routines when I was doing some of my undergraduate work and even my graduate work. Some basic understanding of mathematics and electronics is probably required to use these routines. I don't think the average computer owner/user will have a need for these routines. However, the serious electronics hobbyist, student, or designer will find these programs most beneficial.

The students can learn more easily from comparing their answers to those found from the programs. Usually students are limited by only a few examples in text books. These programs enable the student to go beyond just the basic examples. As for the electronics hobbyist or designer, no one really sits down and designs by hand anymore. Sure, the designer has to come up with the idea, but the computer can do all of the tedious job of number crunching (which if done by hand is prone to error). This disk demonstrates that there is more to computing than just system utilities or file processing.

NAME: rc

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$1CB

DESCRIPTION: This routine is a simple Basic09 program that prints out parameters of an RC circuit (resistor-capacitor) as the resistor heats up.

USE: run rc <cr> or rc <cr>

You must modify the initial conditions inside the program itself for each case. Therefore, it would not be wise to pack this program. It would be cleaner to pass the initial conditions as variables allowing this program to be a stand-alone module. The initial conditions are: change, time, resistance, capacitance and voltage.

AUTHOR: George Dorner

NAME: ResRatIHC

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$176

DESCRIPTION: This routine finds resistors for a specified ratio and specified percentage tolerance limits.

USE: resrat1hc <cr> self prompting

AUTHOR: Carl R. Kreider

NAME: network

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$114E

DESCRIPTION: Electronic circuit network design/analysis tool. There was one error in the program that is easily fixed. Change:

```
IF HCS="Y" OR HCS="y" THEN
    OPEN #outpath,"/P1"
```

to:

```
IF HCS="Y" OR HCS="y" THEN
    OPEN #outpath,"/
P1":WRITE
```

NOTE: /P1 could be /P depending on what your printer driver is set to
USE: network <cr> the program prompts you for a filename if you have a previously saved ".net" file or for manual data entry. You have the option of editing data points but not of saving them from this program. I like the feature of using a data file. One cardinal rule about computers and data, you should never have to enter the same data into a computer by hand more than once. This feature can change any program into a very powerful routine. I did notice that this program will not evaluate DC circuits so don't try using a frequency of zero hertz! The output (either CRT or hardcopy) gives you output frequency in hertz, gain in decibels, and the associated phase angle in degrees. You have a choice of either a linear scale or a log scale for frequencies. This is a great tool for designing filters and such. It handles passive as well as active devices such as resistors, capacitors, inductors, field effect transistors, NPN transistors, and operational amplifiers. The help file "network-how" is very comprehensive. Several example ".net" files are included to provide you with extra help in understanding how the program works.

AUTHOR: Carl R. Kreider

Based on article in EDN Magazine; schneider 10/5/27, Niemeyer 2.4.81, and Steincross 9/1/82.

NAME: fast_fourier

TYPE: Math, data analysis/package

LANGUAGE: Basic09

SIZE: \$16FE

DESCRIPTION: This algorithm performs fast fourier transforms and inverse transforms on data (number of samples must be a power of two). Many periodic functions such as square waves or triangular waves are really just a combination of sinusoidal terms. The square wave for example is derived by adding all of the odd harmonics of a sine wave together. The techniques used today to represent such functions was first presented in 1822 by a famous French mathematician and physicist Jean-Baptiste Joseph Fourier. Today these mathematical series are known as Fourier series. The transform is a technique for going from data mapped in the frequency domain to the time domain. You can also go back from the time domain to the frequency domain.

USE: fast_fourier <cr> menu driven you have the option of loading in a file or entering data in by hand. Another neat feature of this program allows you to save your data to a file!

AUTHOR: Greg Morse

NAME: linefit

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$1060

DESCRIPTION: Provides you with three equations that each minimize the X distance, the Y distance, and the perpendicular distance from each data point to a line.

USE: linefit <cr> all data must be entered by hand

AUTHOR: Greg Morse

NAME: normal

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$241

DESCRIPTION: Demonstrates how to generate random numbers which have a gaussian distribution with a chosen mean and standard deviation.

USE: normal <cr> uses two programs rnd_smp1 is an important routine that generates sample data for a normal population. Rnd_smp1 was derived from "Simulating Sampling from Normal Populations" by Catherine Lilly and Two-Year College Mathematics Readings; Mathematical Association of America, 1981

AUTHOR: George Dorner

NAME: univariate

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$6CF

DESCRIPTION: Statistics Package provides standard deviation, mean, median, minimum and maximum values, range, average deviation, variance, coefficient variance, standard error, skewness, and kurtosis. It also provides population parameters.

USE: univariate <cr> prompts you for input — manual data entry

AUTHOR: George Dorner

NAME: stdev

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$2D3

DESCRIPTION: Subroutine, requires a main program to call it. Calculates statistical values of interest for two arrays of up to 2000 points. Provides standard deviation.

USE:

argvar (xsiz,ysiz,x(i),y(i),statres)
xsiz and ysiz are sizes of array

Continued on Page 20

SPECIFICATION FOR SERIAL INPUT/OUTPUT DRIVERS FOR THE TANDY COLOR COMPUTER AND THE OS-9 OPERATING SYSTEM

Downloaded from CompuServe

Section 1.0: Introduction

All of the Tandy supplied serial drivers for the Color Computer are optimized for driving remote and local terminals to the Color Computer "Console". These drivers are also suitable for general purpose serial input-output drivers, but are not optimized in that application. In fact SCFMan itself assumes a terminal and human operator on the acia port(s).

This document is intended to serve as a discussion instrument and tool in deriving a new class of driver for the machine. A prototype driver exists and is found in the DL 7, (database), of the Compuserve OS-9 special interest group, (SIG). This driver was written to implement the Terminal Communications Package, BiTerm, and is identified as "S10". As this document is being created "on-line", any member of the OS-9 community should feel free to input comments, suggestions, and suggestions to the author: Wm. L. Brady, 70126, 267, 4776-B Carmody Court, Harwood Md. 20776, (301) 952-1761, who assumes the maintenance responsibility.

Section 2.0: Applicable documents

OS-9 Technical Information Manual and addendums Microware Systems/Motorola/Tandy Corporation(s).

Section 3.0:

Preliminary Requirements

1. The driver must conform to all applicable OS-9 conventions and standards extant for device drivers, in regards to position independent code, uses of the interrupt facility and device descriptor format, including non-fixed device addresses. However, the driver shall not, necessarily, mimic the operation of all bytes in the device descriptor format, and may re-define any byte that is not required for correct execution of system modules above the driver level, (SCFMan).
2. The driver shall use interrupts to signal the system and user software of each ASCII character received, except for, (when available), DMA, (direct memory access), devices in which case the interrupt shall be generated when the port hardware requires service by the user software. For DMA devices the interrupt shall be user selectable as to generation after timeout, (user specified), or buffer full.

3. The receive buffer shall be user selectable, and shall not be less than 64 characters nor more than 2048 characters.
4. The Transmitter function of the driver may or may not be interrupt driven, however, if interrupt driven shall use a separate interrupt handler, loaded into the OS9 IRQ interrupt facility at a lower priority than the receiver interrupt, and meeting the same buffer requirements as 3 above.
5. The driver shall generate user definable, (via the device descriptor, x-off and x-on), flow control characters TOWARDS the host, and shall not respond to the same characters when received FROM the host.

a) At the users option, via a setstat system call, the receiver shall generate an x-off character whenever a user specified character, (usually SOD, carriage return), is received by the host.

b) At the users option, via a setstat system call, the receiver shall generate an x-off character at a user specified buffer count, as specified in the device descriptor.

c) The driver shall automatically de-select method a) if method a) is active and method b) is selected, and vice-versa. The driver shall guarantee that only one mode is active at any given time. The selection of one method without deselecting the other shall not cause an error to be generated by the driver.

d) If the driver has sent an x-off, it shall not resend it, and shall send x-on if either a user settable, (via the device descriptor), buffer count is reached, or if a getstat \$01 is executed or a read is attempted against an empty buffer.

5. The driver shall completely configure the port hardware on the INIT system call. All information used by the driver shall be present in the device descriptor at the time of this call.
6. The driver shall completely re-configure the port, after the INIT system call by the SS.ComST setstat call \$28. All information used by the driver shall be present in the device descriptor at the time of this call.

7. The driver shall provide, as a minimum, three "filter" capabilities. A filter is defined as a criteria which a received byte must meet prior to its being placed in the input buffer by the driver.

a) One filter which shall delete all characters GREATER THAN, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

b) One filter which shall delete all characters LESS THAN, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

c) One filter which shall delete all characters EQUAL TO, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

8. The driver shall monitor detected parity errors and overruns on the port hardware.

a) Overruns shall be processed by the interrupt handler by inserting a user specified, (via device descriptor), graphics character in the input buffer. The driver shall not generate an error for this condition.

b) Parity errors shall be processed as any other character, however, the driver may be

designed to replace any errored character with the same graphics character as in a) above.

9. The driver shall reset its buffer pointers at user request via a `setstat` call. This is to allow the user to "replay" the entire buffer contents for a repeat screen display function or a "quick print feature".

Section 3.A: Optional Driver Features

1. The driver may have the capability to open and send characters directly to the standard output path, (screen), however, this capability shall be controlled by setstat calls, and shall be deselectable. This capability shall not replace any of the capabilities in section 3.0.
2. The driver may have the capability to send characters directly to one or two user specified path(s), (method of specification TBD), however, this path shall not use the primary driver buffer or buffer pointers. This feature shall be controllable by setstat calls, and, if implemented, meet the filter requirements in 3.7 a, b and c above (this feature would support on-line printing and downloads). It is recommended that feature automatically select an x-off method as specified in 3.4 above.

Table 3.0:

Additional Replacements to the "T" class of Device Descriptors

Name	Relative Address	Size (bytes)	Use
IT.FLB	tbd	1	Greater than filter byte
IT.FLL	tbd	1	Less than filter byte
IT.FLE	tbd	1	Equal to filter byte
IT.BAD	tbd	1	Bad Character (overrun) Graphics character
IT.BUF	tbd	1	Buffer size selection (table entry)
IT.EOR	\$1D	1	Same as T class driver but different usage see 3.5.a
IT.MIN	tbd	2	Buffer near empty count for xon
IT.MAX	tbd	2	Buffer near full count for xoff
IT.BF1	tbf	2	Buffer 1 (DMA devices only)
IT.BF2	tbd	2	Buffer 2 (DMA devices only)

35-9

**Don't Miss
RAINBOWfest Princeton
October 17-19, 1986!**



POSITIVE ACTION VS. NEGATIVE COMMENTS

by Dale Puckett, Director-at-Large

We OS-9 users live in a small universe. There aren't too many of us. Sure, our favorite operating system appears to be coming out of the underground and capturing its own spot in the popular press this

year. But, if we all really want to see it succeed and become the MD-DOS equivalent in the 68000 marketplace, we must all do our part.

"What can I do," you ask? Become an OS-9 evangelist.

That's a natural. I'll bet you are already doing it. If I were a fly on the wall, I'd hear you telling all your MS-DOS friends about the many great OS-9 features you have grown to know and love.

OK, we're all marketing OS-9 to the best of our abilities, but, are we covering all bases? I don't think so because we don't have everything we need. If OS-9 is to succeed in the consumer marketplace it must have a reason for being. There must be hundreds of application programs that do jobs that people are tired of doing by hand.

To make this happen we must encourage programmers within our ranks. We must salute them when they move the state of the art forward. We must encourage them to make bold steps forward with innovative techniques rather than discourage them with our criticism.

I say these things after reading the mail on CompuServe's OS-9 SIG and on Delphi's Color Computer SIG. There are a lot of helpful people on both of these electronic bulletin boards, but there is also much criticism. This criticism, especially when combined with low sales and minimal profit, discourages programmers from writing new programs. As a result, we all suffer.

People criticize Tandy for making a business decision when they designed their implementation. Then, they criticize Microware for delivering the product the customer ordered. They don't stop to realize one important basic of the business world — the customer is always right.

Frankly, Tandy had a good reason for every "feature" they put in

the Color Computer implementation of OS-9. We may or may not agree with the selection made by Tandy's designers but we must realize that no business decisions are made lightly. And to be quite honest, we must also realize that if Microware hadn't been willing to deliver the product that Tandy wanted, Tandy probably would have chose another operating system for the Color Computer. Then where would we be?

In the old days the story read, "Those who can, do. Those who can't, teach." If recent experience is any indication that old parable might be reworded ever slightly. "Those who can, do. Those who can't, criticize."

In case you are wondering, this column was born as the result of several interchanges on one of the bulletin boards above. Another use took me to task for being positive and the debate went back and forth for several weeks.

I wasn't sorry for being positive then. I am not sorry now. The world would be a lot nicer place to live if more people had a positive attitude. Because of this belief, I work very hard to maintain and project a positive attitude even though it is very hard at times.

As I told my adversary, everyone deserves a pat on the back every once in awhile — especially when he is not getting rich in the market he is supporting out of pure love.

If this positive attitude theory seems like a lot of hogwash to you, I challenge you to look around the business world. I think you'll find that the companies that project a positive "can do" attitude are the ones that are getting ahead. Organizations run by managers who try to think of reasons not to do something are falling like flies.

The bottom line, if you know a better way to do something, do it. If it's something everyone can use, sell it. If not, share it. But, do it. And, don't put the other guy down just because his approach is a little different.

Application for membership in the OS-9 Users Group

Last: _____

First: _____

Middle: _____

Street: _____

City: _____

State: _____ ZIP: _____

Phone: (_____) _____

Delphi ID: _____

Computer Make and Model: _____

OS-9 Level: _____ (I) _____ (II) _____ (68K) _____ (CoCo)

Disk Size: _____ (5¼") _____ (8")

Disk Format: _____ (CoCo) _____ (Standard)

Single Sided: _____ Double Sided: _____

of tracks: _____ (35) _____ (40) _____ (80)

Other restrictions, formats, comments, etc: _____

Enclose \$25 to cover the first year's membership in the form of a personal check or money order. Please allow approximately 3-5 weeks for processing your membership. The Users Group has a small staff, and applications are averaging 30 to 50 a week.

Shortly after acceptance of your application for membership, you will receive the current Group newsletter ("MOTD"), and soon after, the "starter" diskette, UG Disk # 0, with software of the type useful in getting you started with both OS-9 and the Users group, including a modem program to assist you with CompuServe access while under OS-9. Additional diskettes may be purchased at a cost of \$5 each to cover the cost of media and postage.

Mail your application and other correspondence to:

The OS-9 Users Group
ATTN: (department)
9743 University Avenue
Suite 330
Des Moines, IA 50322

Where (department) is President, MOTD Editor, Librarian, Membership, etc.

Membership dues can be charged to your VISA or MasterCard. Simply enter your card information below and sign this form.

Card type: VISA _____ MasterCard _____

Account #: _____

Expires: _____

Is this a Renewal? Yes _____ No _____

LETTER TO THE EDITOR

Gentlemen;

We have been discussing on the OS-9 SIG a new class of serial I/O driver for OS-9. I have put up a "straw-man" spec in d17. (SSPEC.DOC). You may want to pick it up for publication in MOTD.

Bill Brady

You'll notice SSPEC.DOC in this issue of the MOTD. It's always fun for me to be able to respond to people that have something positive to add to the MOTD. Constructive enhancements to the world of OS-9 are what will help it to grow into THE operating system of the future.



at RainbowFest Princeton
October 17-19, 1986

Meet a Hero (of OS-9)

Get answers

Come see: Dale L. Puckett, Director-at-Large
Brian A. Lantz, President
Bill Turner, Vice President
Bruce Warner, Editor, MOTD

Make friends

Have breakfast Sunday morning with Frank Hogg of
Frank Hogg Laboratory

. . . get the straight answer for what's
coming down the line.

Breakfast Tickets - \$14.00

Get advance tickets by writing:

Share knowledge

Renew membership

OS-9 Users Group
ATTN: Breakfast
Suite #330
9743 University Ave.
Des Moines, Iowa 50322

Sign up a friend

Solve problems

PIPE OF THE MONTH

by Kevin Kuehl

In this article I am going to show you how to build a simple but powerful database manager using only six common utilities. The procedure is very simple and uses only cat, del, grep, rename, sort, and tr in three steps with one being a pipeline.

In my example I will be building a list of addresses in a file called 'addrbook.' The file has this format:

```
lastname, firstname~address~city, state zip~<cr>
```

Step one is to type this statement:

```
OS9: cat > tempfile<cr>
```

```
Kuehl, Kevin~806 Division Road~Valparaiso, IN
```

```
46383~<cr>
<eof>
```

With this I have used cat to put my address in a file named 'tempfile.' But why are the tildes in the line? The pipeline needs some character to serve as a separator which will become a carriage return when run through. Why have I used the tilde? Because I have yet to see it in an address. Actually the separator can be any character you want, but I have found the tilde to work best. Why have I included a tilde at the end of the line when tr will print the last carriage return? If you have a number of lines that the search string matches, the extra tildes add

one blank line in between them. So really the only reason it is there is for aesthetics.

The next step is to append this information to the end of the file 'addrbook.' If this file doesn't already exist, then you can simply change the previous command to send its output to that file. Otherwise you should use cat, del and rename like so:

```
OS9: cat addrbook tempfile >
newfile; del addrbook
tempfile; rename newfile addrbook<cr>
```

This updates 'addrbook' to include the new information included in 'tempfile.' In actual use, I would keep this line in a shell script with a simple name such as update. If you have one, a program such as cp which appends to the end of a file if it already exists is a time saver. I have one and I just type.

```
OS9: cp tempfile addrbook<cr>
```

and the computer does the work. Also 'tempfile' can include any number of new additions — not just one as in my example.

The final step is to get a printable output given a search string. My address can be found by typing:

```
OS9: grep Kevin addrbook | grep
Kuehl | tr~ /13<cr>
```

Why did I use grep twice? With

only one I must search for either 'Kuehl' or 'Kevin,' but not both. This way I am able to make sure I get my name only. If I were to search for 'Kuehl, Kevin,' grep would become confused and look for 'Kuehl,' in the file 'Kevin' — definitely not what we want. The first grep matches all 'Kevins' and the second matches only 'Kevin Kuehls' then sends its output to tr which changes all the tildes to carriage returns and gives us:

```
Kuehl, Kevin<cr>
806 Division Road<cr>
Valparaiso, IN 46383<cr>
<cr>
```

Why did I put the name at the beginning of the line last name — first, first name — last? So that a sort program would work on the name correctly. If I were to type:

```
OS9: sort addrbook | grep IN |
tr~ /13<cr>
```

I would be presented with a neatly sorted listing of all the people in 'addrbook' who live in Indiana.

I hope this short article has helped you put to use some common utilities and maybe even given you some ideas for others. The procedure I have laid out should work with any set of utilities since I have not used any options except for the decimal representation of a carriage return in tr.

PATCHING STRIP

by Jay Truesdale

Here is a copy of a message I posted in reply to Dave K. about the strip utility:

Dave, in regards to my previous message about the strip utility. The problem with the "meaningless characters" appears to be fixed by changing the LDX #CHAR to LEAX CHAR,U as I suggested.

I believe the "meaningless characters" are actually control characters, which in my case, cause my WordPak II to perform functions like cursor addressing, etc. The description of the I\$Write system call in my (Color Computer) OS-9 manual says that register X is to contain the starting address of the data string to

be written. From looking at the object code generated by the assembler, it looks to me like the instructions LDX #CHAR would load register X with the value \$000D. The following I\$Write system call would then proceed to write the two bytes that are at location \$000D, which could be just about anything, and in this case are probably seen as control characters by my WordPak II.

After reading this, I realized it would make more sense if you read Dave's letter; it is on the OS-9 SIG on Subtopic 3.

I hope this is correct and makes sense, if not please let me know.

PLACING AN AD IN THE MOTD

There are still a lot of people out there that want to advertise in the MOTD. So what do they do?

Well, here's the deal! You start out by getting your ad made up (called camera ready copy). When that is completed, you submit your ad with a check made out to the OS-9 Users Group and send both to:

Editor, the MOTD
c/o Bruce N. Warner
14503 Fullerton Road
Dale City, VA 22193-2034

The price for advertising in the MOTD is based on the size, location and number of colors (colors limited to black and red). This chart explains more completely.

SIZE	REGULAR		SPECIAL (back cover)	
	1-Color	2-Colors	1-Color	2-Colors
Full Page	\$400	\$480	\$500	\$600
Half Page	\$200	\$240	\$250	\$300
Quarter Page	\$100	\$120	\$125	\$150
Eighth Page	\$ 50	\$ 60	\$ 62.50	\$ 75

Rates are higher for the special issue that will be inserted in RAINBOW magazine. These rates will be published at a later date.

You'll have to make sure that you've included your camera ready copy and a check for payment made out to the OS-9 Users Group. The deadline for entries is the first of every month for the issue beginning the following month.

A PATCH FOR MORE STORAGE

by Steve Bjork and Pete Lyall

(Editor's Note: The following information is provided courtesy of CompuServe.)

Getting more out of (or perhaps on to) the DeskMate disk can be as easy as making a few patches to the CCDISK module. Here are the necessary offsets and values.

Offset	Old	New	
1f9	13	10	6 ms step rate
1ff	22	02	Shorten commands delay
209	01	01	Drive 0 (front of DS 0)
20a	02	41	Drive 1 (back of DS 0)
20b	04	02	Drive 2 (front of DS 1)
20c	40	42	Drive 3 (back of DS 1)
2e8	03	00	6 ms restore
341	60	84	New CRC value
342	da	df	New CRC value
343	8c	79	New CRC value

MY CADILLAC FOR THE COCO

by Richard Johnson, CompuServe OS-9 SIG

This driver was written out of a need to support the extra function keys on my keyboard. I have the "Deluxe CoCo" keyboard which has a true control key, an alt key, and an F1 and F2 key. It has the arrow keys arranged in a cluster and has the clear key positioned by the '@' key. I realized that PBJ wasn't going to support non-standard keyboards, so I wrote this driver. It will support any keyboard that has extra function keys mapped into the four extra slots in the standard keyboard's matrix. It also allows all keys to be reprogrammed to return new values.

Any value may be returned except \$FF and \$FD, which is used by the driver internally, \$FE, which is used for the Caps Toggle value instead of the \$1F that both the standard CCIO and WordPak drivers use. I changed it so that the keyboard could send all ASCII values (\$00 through \$7F). This allows the keyboard to emulate any keyboard that

sends single character codes for key depressions. Also, those who want a DVORJAK keyboard layout can have it just by reprogramming the key tables. As released the keyboard works exactly as before with the following exceptions:

the CLEAR key sends a \$08 (or Backspace) in both the normal and shifted modes. Ctrl-Clear sends a rub-out (\$7F).

the CTRL works as the control key the ALT is now an escape key and sends \$1B in all modes

the F1 key sends a (unshifted) a '\', (shifted) a '[', and (ctrl) a ']'.

the F2 key sends a (unshifted) a '~', (shifted) a '^', and (ctrl) a 'J'.

I chose these for convenience in 'C' programming. Using the reprogramming option these keys as well as 1 through 0 and :-@ plus the <BREAK> and <Spacebar> keys can be changed. Below is a chart that shows the positions of all the keys in the Keyboard matrix.

KEYBOARD TABLE
COCO deluxe keyboard matrix

		A	B	C	D	E	F	G
1	H	I	J	K	L	M	N	O
16	P	Q	R	S	T	U	V	W
32	X	Y	Z	ua	da	la	ra	ap
48	#	1	2	3	4	5	6	7
64	@	:	;	'	-	=	/	
80	en	cl	br	ctr	alt	f1	f2	sh

** ----- used for the joysticks -----
1 2 3 4 5 6 7

LEGEND :

ua = uparrow	en = enter
da = downarrow	cl = clear
la = left arrow	br = break
ra = right arrow	ctr = control
sp = space bar	f1 = function key 1
sh = shift	f2 = function key 2

How To Reprogram Keys

The keys are arranged as in the Matrix in the keyboard table with the '@' key followed by the <ua> (uparrow) through <f2> keys. There are three tables, each containing a slot for each key. To find a key in any table use the locations in the above table. The keys may be reprogrammed by using OS-9's display command or by SetStat calls. A key's current value may be obtained by a GetStat call. The Control and shift keys can only be changed in the driver source code and changing their values in the tables will have no affect on them.

There are three Display codes, each corresponding to one of three tables. The first is \$1D for the

Unshifted keys, the second is \$1E for the Shifted keys and finally \$1F for the Control keys. The use of the Display command:

Display 1F {value} {key offset}

Where "\$1F" could be either \$1F, \$1E or \$1D, for the respective table; "value" is the new value of the key and "key offset" is the offset of the key into its table. All values must be in Hexidecimal and be of one byte in length.

Using the Status Calls works in similar ways except the codes for each table are 150 unshifted, 151 for shifted and 152 for control. The use of both the SetStat and GetStat calls is described below.

For Setstat -

Entry

A = Code (15# dec)
B = New Key Value
X = offset into the Table

Returns

B = B
cc is clear

A GetStat for the Function keys is also included -

Entry

A = Code (15# dec)
X = Offset into table

Returns

on success -

A = key value
cc is clear

on error -

B = error code
cc is set

Compatibility With Other Drivers

This Driver, WordPakC, only works in a Radio Shack compatible mode. Some O-Pak compatibility has been implemented. It is not compatible with the standard WordPak mode. All the functions of the Radio Shack CCIO that relate to the text screen have been implemented and the WordPak driver 3.1 RS functions have also been added. The last page is a chart showing display codes and their effects. The cursor may be changed either by using sequences or by the RS compatible "05 nn" codes. Note however that since the WordPak is only a monochrome display board the shape and blink rate are changed instead of the color.

Status Calls (Getstat) -

All of the CCIO codes are supported (including joysticks) except the following two — SS.DSTAT, which returns graphic screen data, and SS.ALFAS, which is dependent

on a memory mapped display. The following O-Pak calls are supported as well —

CODE 131 — last displayable character — always returns \$7F in A.

CODE 132 — O-Pak screen size call. Don't use this call, included for software compatibility. A = columns, B = rows on screen (80,24).

CODE 134 — Cursor Status. A = cursor blink rate, B = cursor type (Block or UL)

CODE 135 — Overstrike Mode — always in overstrike mode. A = 0.

As the standard CCIO is never likely to support these calls, I have only included them for completeness. I recommend that you never use them in your own programs to maintain compatibility with CCIO.

How to Install WordPakC and Term

First assemble the files WordPakC.asm and TermC.asm into executable files. Use the following line:

OS9: asm WordPakC.asm o=/d0/WordPakC #30k;asm TermC.asm o=/d0/TermC #20k

If you have any WordPak drivers, just use their install procedure, except substitute WordPakC for WordPak in the file Bootfile. If you have Sdisk, do the same for CCIO and use the TermC descriptor Term for WordPakC. (TermC will appear as Term when loaded so no patching of Init is needed and new BootFiles may be created using it; TermC is exactly like the PBJ Term.)

If you would like a Write only version to use along with CCIO, you can carefully delete all keyboard functions, the keyboard table, the Getstat and Setstat Codes for key-

board use and the display code for the Function Key changing and make the driver a write only driver. Then use the following for Read and Setstat:

```
READ  
PUTSTAT  
clrb  
rts
```

And the driver will only be capable of display. This will eliminate the dread keyboard lock up that comes from trying to use both the PBJ drivers and CCIO at the same time. Change the name of TermC from "Term" to "WP" so that it doesn't conflict with the CCIO's Term.

I have done the best job I could on this driver, however, I'm not the best assembly language pro-

Continued on Page 12



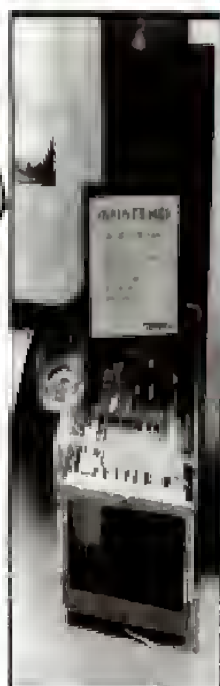
Right: Bruce Campbell, President of Intel USA, Inc., demonstrates the latest version of Sculptor (The Fourth Generation Programming System) in the Glens Booth.



Right: OS-9 running on an IBM PC compatible machine. The TLM 6000 OS-9 board is an IBM XT showing that the IBM line have more uses than keeping the floor open.



Above: The man in the jacket is laughing because everyone thinks you have to have an IBM PC to get anything done. In fact, he's using an Apple IIe with the MicroTRENDS implementation of Jonathan. This could be the start of a real computer. Top Right: Virtually unlimited power is possible in a small case. At the Microware booth, the CMX-400 single-board computer was displayed to point out the possibilities of having real computer power in a limited office space environment. Bottom Right: If you want the TLM 6000 OS-9 board, you might have never see Bill Moore at the IBM PC keyboard, but innovation can change almost anything.



WORLD COMPUTER CONFERENCE by Bruce Warner

LAS VEGAS — If there's one thing you can say about the National Computer Conference here, it's that OS-9 was one of the VERY few new items on the agenda. It took as little rest of the industry was out to lunch while OS-9 was taking the leading edge of which new.

For those of us that read the MOTO, there's nothing new about CD-ROM, but for the many attendees at the conference, CD-ROM was absolutely new and exciting. My greatest joy was watching at the MS-DOS machines that failed to meet the abilities of OS-9, even OS/2 Level 1 OS-9.

Seeing OS-9 in the Microware booth was no surprise, but when I looked in some of the other booths and found OS-9 alive and well, I was elated.

Glens, an Old Hand at OS-9

The Glens booth was displaying OS-9 on their 68000 machines, including a new 68000 single-board computer that fits in the case of an IBM PC. While it's not an identical footprint of the IBM motherboard, it is a well-designed computer that allows you all the capabilities of OS-9 without the need for an expensive, custom-designed power supply and case. The Taiwan specials were custom-made for this little jewel.

Slow Special Carried to US From Data-Comp

One of the specials of the show has been extended to the Users Group through October 1. Thanks to Don Williams of Data-Comp, you can order the Mustang420 — 100 Golden Eagle Special through October 1, 1986 at just \$3,995.00. This includes the following features:

- CMX Micro-20MCS500 SBC (for the 3MC 68000 15.67 Mhz CPU and 500K)
- DATA-COMP Metal Enclosure with switching power supply
- DS, DD, 80 Track Floppy Disk Drive
- 25 MByte Winchester Hard Disk Drive
- XEBEC 1410A Hard Disk Controller

- All Special Cables-Factory Assembled 45 Hour Burn-in
- Choice of Operating System: OS-9, UniFLEX
- Both the OS-9 and UniFLEX operating systems are UNIX System V, C compatible
- 4 Serial Ports, RS-232C

Don also informs me that you must make payment in full with your order to qualify for this discount. Add to this an additional 10 to 20 percent discount on all EXX software to registered owners. You can reach Data-Comp at 5800 Castaneda South Road, Houston, TX 77040 or call at (615) 342-4600.

Sculptor Keeps Getting Better

Another novelty at the show was the update of Sculptor. It's now faster than ever, and still lives up to all its previous flexibility. The greatest increase is in its ability to execute statements without having to reload a shell for each execution. This alone greatly enhanced the speed of Sculptor.

CD-ROM Conference

The CD-ROM conference was another session worth attending. This one really put us in beyond the big boys. The funniest question came from an attendee that asked what would happen if Big Blue were to try to get involved. Logic dictates the answer to this one. Big Blue is not a developer of operating systems. Neither are they the hardware developers of compact disk players. The story of CD-ROM lies between these two marketers, and Big Blue is not in the drivers seat. Facts are facts, and CD-ROM is based on OS-9.

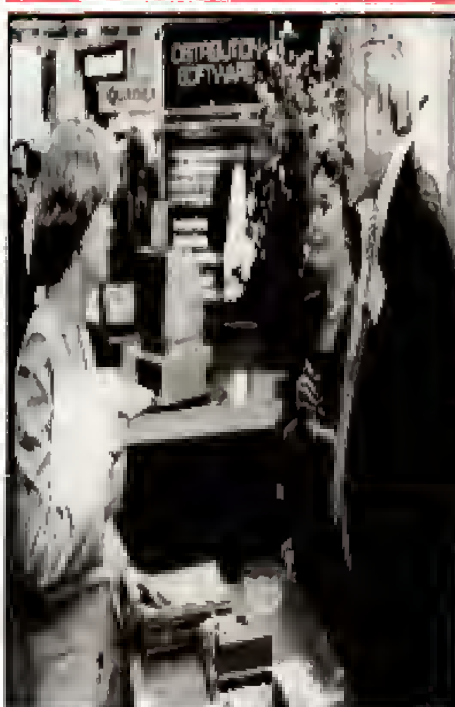
The balance of the show was fairly bland, leaving most of us a little cold. I was somewhat disappointed with the overall show. Luckily, I wasn't the only one to have these feelings. I was more than a little happy to repack my bags and head for the airport.



Above: There's almost no limit to the number of 68000 OS-9 computers on the market. As DATA-COMP showed off their implementation of the CMX-400 in the Glens booth. Left: Microware founder Ben Kaplan speaks to the many visitors at the CD-ROM press conference. For those of us who have followed the story of CD-ROM to the MOTO there wasn't much new to hear about, but in the world as a whole, CD-ROM was news for sure.



Left: The complete Caliber Encapsulation is accessible from a single compact disk. This is the foreboding of the industry shattering CD-ROM. The reason presented in the Philips booth at SCC was running on the Apple IIe.



An OS-9 laptop (LJ) talks with Jerome Kaplan and Bill Moore about the advantages of OS-9. (The visitor had her back pointing directly at the IBM booth).



The case looks a lot like the whole you see from Big Blue, don't you? Inside this case lies the heart of a real single-board computer.



Above: Is there any link to the possibilities of OS-9? Evidently not, as OS-9 was running on the Motorola System 1000. Right: While the Atari ST was raising Heck, Bill Moore decided to use another task on the remote terminal to demonstrate the ability to show visitors. Eye Alvin gets a chance to grow out of the game world when you offer them OS-9 as an operating system.



Right: Bruce Campbell, President of mpd USA, Inc., demonstrates the latest version of Sculptor (The Fourth Generation Programming System) in the Gimix Booth.



Right: OS-9 running on an IBM PC could make the PC a real computer. The TLM 68000 OS-9 board in an IBM XT shows that the IBM does have more uses than keeping the door open.



Above: The man in the jacket is laughing because everyone thinks you have to have an IBM PC to get anything done. In fact, he's using an Apple IIe with the MicroTRENDS implementation of Jonathan. This could be the start of a real computer. Top Right: Virtually unlimited power is possible inside a small case. At the Microware booth, the GMX-020 single-board computer was displayed to point out the possibilities of having real computer power in a limited office space environment. Bottom Right: If it weren't for TLM's 68000 OS-9 board, you might have never seen Bill Moore at an IBM PC keyboard, but innovation can change almost anything.



NATIONAL COMPUTER CONVENTION

by Bruce Warner

LAS VEGAS — If there's one thing you can say about the National Computer Conference here, it's that OS-9 was one of the VERY few new items on the agenda. It looks as if the rest of the industry was out to lunch while OS-9 was taking the leading edge of what's new.

For those of us that read the MOTD, there's nothing new about CD-I ROM, but for the many attendees at the conference, CD-I ROM was absolutely new and innovative. My greatest joy was laughing at the MS-DOS machines that failed to meet the abilities of OS-9, even 68009 Level I OS-9.

Seeing OS-9 in the Microware booth was no surprise, but when I looked in some of the other booths and found OS-9 alive and well, I was elated.

Gimix, an Old Hand at OS-9

The Gimix booth was displaying OS-9 on their 68000 machines, including a new 68000 single board computer that fits in the case of an IBM-PC. While it's not an identical footprint of the IBM motherboard, it is a well-designed computer that allows you all the capabilities of OS-9 without the need for an expensive, custom-designed power supply and case. The Taiwan specials were custom-made for this little jewel.

Show Special Carried to US

From Data-Comp
One of the specials of the show has been extended to the Users Group through October 1. Thanks to Don Williams of Data-Comp, you can order the Mustang-020 — 100 Golden Eagle Special through October 1, 1985 at just \$3,998.80. This includes the following features:

- GMX Micro-20MC68020 SBC (for the MC 68020 16.67 MHz CPU add \$300)
- DATA-COMP Metal Enclosure with switching power supply
- DS, DD, 80 Track Floppy Disk Drive
- 25 MByte Winchester Hard Disk Drive
- XEBEC 1410A Hard Disk Controller

- All Special Cables-Factory Assembled-48 Hour Burn-in
- Choice of Operating System. (OS-9, UniFLEX) (Both the OS-9 and UniFLEX operating systems are UNIX System V, C level compatible.)
- 4 Serial Ports, Pre-Wired

Don also informs me that you must make payment in full with your order to qualify for this discount. Add to this an additional 10 to 20 percent discount on all 68XXX software to registered owners. You can reach Data-Comp at 5900 Cassandra Smith Road, Hixson, TN 37343 or call at (615) 842-4600.

Sculptor Keeps Getting Better

Another novelty at the show was the update of Sculptor. It's now faster than ever, and still lives up to all its previous flexibility. The greatest increase is in its ability to execute statements without having to reload a shell for each execution. This alone greatly enhanced the speed of Sculptor.

CD-I Conference

The CD-I ROM conference was another session worth attending. This one really put us in beyond the big boys. The funniest question came from an attendee that asked what would happen if Big Blue were to try to get involved. Logic dictates the answer to this one. Big Blue is not a developer of operating systems. Neither are they the hardware developers of compact disk players. The story of CD-I ROM lies between these two marketers, and Big Blue is not in the drivers seat. Facts are facts, and CD-I ROM is based on OS-9.

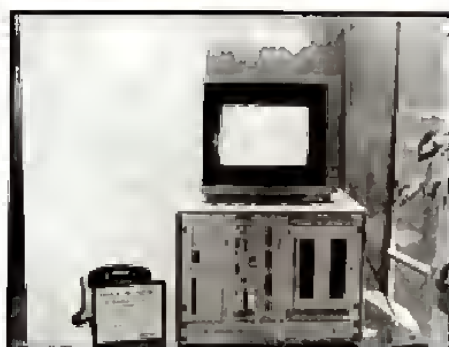
The balance of the show was fairly bland, leaving most of us a little cold. I was somewhat disappointed with the overall show. Luckily, I wasn't the only one to have these feelings. I was more than a little happy to repack my bags and head for the airport.



An OS-9 hopeful (L.) talks with Jeanne Kaplan and Bill Moore about the advantages of OS-9. (The visitor had her back pointing directly at the IBM booth.)



If the case looks a lot like what you see from Big Blue, don't fret. Inside this case lies the heart of a real single-board computer.



Above: Is there any limit to the possibilities of OS-9? Evidently not, as OS-9 was running on the Motorola System 1000. Right: While the Atari ST was running Hack, Bill Moore decided to run another task on the remote terminal to demonstrate its flexibility to a show visitor. Even Atari gets a chance to grow out of the game world when you offer them OS-9 as an operating system.



Above: There's almost no limit to the number of 68020 OS-9 computers on the market, as DATA-COMP showed off their implementation of the GMX-020 in the Gimix booth. Left: Microware founder Ken Kaplan speaks to the many visitors at the CD-I ROM press conference. For those of us who have followed the story of CD-I ROM in the MOTD there wasn't much new to hear about, but to the world as a whole, CD-I was news for sure.

Left: The complete Collier Encyclopedia is accessible from a single compact disk. This is the forerunner of the industry-shattering CD-I ROM. The version presented in the Philips booth at NCC was running on the Apple IIe.

COCO CADILLAC

Continued from Page 9

grammer and I'm sure that this Driver could stand improvement. So if you can improve it go ahead and do it! Then kindly drop me a

note on the OS-9 SIG on Compu-Serve. My number is 74116,3201 so I can see what you've done. Also be sure to leave a copy in the library.

PO #

SCREEN DISPLAY CODES

Hex	Decimal	Screen Function
A1	A1	Home Cursor
A2 X,Y	A2 X,Y	Position Cursor (X = Column + 32, Y = Row + 32)
A3	A3	Erase Line
A4	A4	Erase to End of Line
A5 n	A5 n	Change Cursor (n = 12 to 42) (Operates differently, See Text)
A6	A6	Cursor Right
A7	A7	Ring Bell
A8	A8	Cursor Left
A9	A9	Cursor Up
AA	AA	Cursor Down
AB	AB	Erase to End of Screen
AC	AC	Home Cursor and Clear Screen
AD	AD	Carriage return
AE 41	27 65	Erase to End of Line
AF 42	27 66	Erase to End of Screen
B0 43	27 69	Insert a Line
B1 46	27 78	Delete a Line
B2 47	27 71	Shift Right
B3 48	27 72	Shift Left
B4 33 3F	27 63 33	Set Normal Video
B5 33 21	27 63 33	Set Inverse Video
B6 36	27 66	Block Cursor
B7 37 r	27 67 r	Cursor Blink Rate (20 = steady, 21 = slow, 22 = Fast, 23 = Invisible)
B8 76	27 118	Underline Cursor
B9 v o	29 v o	Change key at offset 'n' in the unshift key table to value in 'v'
BE v o	38 v o	Change key at offset 'o' in the shifted key table to value in 'v'
BD v o	29 v o	Change key at offset 'o' in the control key table to value in 'v'

(*) - Indicates non CCI0 / RS-089 1.01.00 compatible codes
(**) - Indicates unique to this driver only.

This driver is for Educational purposes and is not to be distributed or

sold as part of any commercial package.

0001		NAM	WORDPAK REPLACEMENT DRIVER
0002		TTL	DEFINITIONS
0003	*		
0004	*		WORDPAK DRIVER
0005	*		
0006		IFP1	ENDC
0007			
0008			
0009			
0010			
0011			
0012	0012	TYPE	SET DRIVER OBJECT
0013	0013	REVS	SET REENT+1
0014	0014	INP12	SET 780
0015			
0016	0016	WORD	WORDP1, WPNAM, TYPE, REVS, ENTRY, MEMSIZE
0017	0017	ORC	V SET
0018			
0019	0019	XPORT	RMS 2 LOCATION OF KEYBOARD
0020	0020	COLCHT	RMS 1 COUNTER FOR KEYBOARD COLUMNS
0021	0021	KEY	RMS 1 CURRENT KEY DOWN
0022	0022	LASTKY	RMS 1 LAST KEY DOWN
0023	0023	KEYVAL	RMS 1 VALUE OF LAST KEY DOWN
0024	0024	CAPLOC	RMS 1 CAPLOCK FLAG
0025	0025	SHFLAG	RMS 1 SHIFT FLAG
0026	0026	CTRFLG	RMS 1 CONTROL KEY FLAG
0027	0027	BUFIN	RMS 1 POINTER TO BUFFER NEXT IN
0028	0028	BUFOUT	RMS 1 ANOTHER FOR READ TO FIND KEY
0029	0029	REPSPD	RMS 1 SPEED OF REPEAT
0030	0030	REPWAT	RMS 1 WAIT COUNT
0031			
0032	0032	ESCONT	RMS 1 COUNTER FOR KEYS IN VIDEO 28
0033	0033	ESCFLG	RMS 1 FLAG THAT HOLD VIDEO ESC CHAN
0034	0034	ESCBUF	RMS 1 HOLDS AND CHAN IN ESC SEQ
0035	0035	CURLIN	RMS 1 CURSOR LINE
0036	0036	CURCOL	RMS 1 CURSOR COLUMN POS
0037	0037	INVFLG	RMS 1 INVERSE VIDEO FLAG
0038	0038	DSTART	RMS 2 DISPLAY START
0039	0039	WORK	RMS 2 A PLACE FOR SHORT TERM STORA
0040	0040	INPBUF	RMS INP12 INPUT BUFFER (IN CASE OF OVE
0041	0041	MEMSIZE	EQ0
0042			
0043			
0044	0044	PCB	READ, WRITE, EXEC.
0045	0045	TTL	WORDPAK REPLACEMENT DRIVER
0046	0046	PCB	"WordPak"
0047	0047	PCB	1
0048			
0049	0049	ENTRY	
0050	0050	LEAD	INIT
0051	0051	LEAD	READ
0052	0052	LEAD	WRITE
0053	0053	LEAD	GETSTAT
0054	0054	LEAD	PUTSTAT
0055	0055	LEAD	TERM
0056	0056	INIT	PERM DP,CC SAVE LAST DP,CC
0057	0057	OROC	FS5J DISABLE INTERRUPTS
0058	0058	CLRA	SET BP
0059	0059	TFR	A, DP TO SYSTEM PAGE
0060	0060		LOOK AS ON9DEFS FOR EXPLANATION OF D. VARIABLES
0061	0061	STU	D.XDDSTA SAVE STORAGE ARE IN XB STOR
0062	0062	LIX	D.IRQ MOVE IRQ ROUTINE TO
0063	0063	STX	D.ALTIRQ ALT STORAGE
0064	0064	LEAX	POLKEY,PC GET LOCATION OF KEY SCAN ROU
0065	0065	STX	D.IRQ READ KEYS AT EVERY INTERRUPT
0066	0066	LEAX	487780 LOCATION OF KEYBOARD PIA

0067	0067	APC610	STX	KPORT,U	PUT IT INTO VARIABLE AREA
0068	0068	5F	CLRB		
0069	0069	APC610	STB	CAPLOC,U	NORMAL MODE
0070	0070	APC610	STA	SHFLAG,U	SET SHIFT OFF
0071	0071	APC610	STA	CTRFLG,U	CLEAR CONTROL FLAG
0072	0072	APC610	STD	BUFIN,U	ZERO BUFFER IN AND OUT POINT
0073	0073	APC610	STA	PI,X	SET UP KEYBOARD PIA
0074	0074	APC610	STA	PI,X	
0075	0075	APC610	STA	PI,X	
0076	0076				
0077	0077	APC610	STD	DSHFT,U	START TO TOP OF SCREEN MEMORY
0078	0078	APC610	STD	ESCONT,U	ESCAPE VARIABLES TO ZERO
0079	0079	APC610	STD	ESCBUF,U	
0080	0080	APC610	STA	CURCOL,U	START OF LINE
0081	0081	APC610	STA	INVTGL,U	NORMAL VIDEO
0082	0082	APC610	COMB		
0083	0083	APC610	STB	PI,X	SET KEYBOARD PIA
0084	0084	APC610	STB	KEY,U	SET KEY BUFFERS
0085	0085	APC610	STB	LASTKY,U	
0086	0086	APC610	STB	KEYVAL,U	
0087	0087	APC610	LDA	PI34	MAKE KEYBOARD REGISTERS
0088	0088	APC610	STA	PI,X	TO R/W
0089	0089	APC610	LDA	PI3F	INSTEAD OF DATA DIRECTION
0090	0090	APC610	STA	PI,X	
0091	0091	APC610	LDA	PI,X	
0092	0092	APC610	CLRA		
0093	0093	APC610	LEAX	WPITBL,PC	INITIALIZE CRTC
0094	0094	APC610	LDB	X+	GET VALUE
0095	0095	APC610	STD	[1,0]	STORE INTO CRTC
0096	0096	APC610	ENCA		
0097	0097	APC610	CMFA	PI14	UNTIL ALL DONE
0098	0098	APC610	BOS	WPINIT	LOOP
0099	0099	APC610	LEGR	WOMCLS	CLEAR/START FRESH
0100	0100	APC610	LDD	PI6400	BLOCK CURSOR, SLOW BLINK
0101	0101	APC610	STD	CURST,PCR	
0102	0102	W	LDA	PIF23	INITIALIZE PORT FOR BELL
0103	0103	APC610	ANDBA	PI11111PI1	SET UP SINGLE BIT SOUND
0104	0104	W	STA	PIF23	
0105	0105	W	LDB	PIF22	
0106	0106	APC610	ORB	PIPIPIPIPIPI	
0107	0107	W	STB	PIF22	
0108	0108	W	ORA	PIPIPIPIPIPI	
0109	0109	W	STA	PIF22	
0110	0110	APC610	CLRB		
0111	0111	APC610	CLRA		
0112	0112	APC610	FULB	PC,DP,CC	OR-STORE AND RETURN
0113	0113				
0114	0114				
0115	0115				
0116	0116				
0117	0117				
0118	0118				
0119	0119	APC610	PIF50573C	WPITBL	PCB
0120	0120	APC610	ICPI5181A	PCB	24,5,24,24
0121	0121	APC610	PIF4	PCB	170,0
0122	0122	APC610	PIF6	PCB	540,0
0123	0123	APC610	PIF6	PCB	PI,PI,PI,PI
0124	0124	APC610	PIF6	PCB	PI,PI,PI,PI
0125	0125				
0126	0126				
0127	0127				
0128	0128				
0129	0129				
0130	0130				
0131	0131	APC610	PIF6	GETSTAT	CMFA
0132	0132	APC610	PIF6	GETSTAT	BNZ
0133	0133	APC610	PIF6	GETSTAT	CMFA
0134	0134	APC610	PIF6	GETSTAT	BNZ
0135	0135	APC610	PIF6	GETSTAT	CMFA
0136	0136	APC610	PIF6	GETSTAT	BNZ
0137	0137	APC610	PIF6	GETSTAT	CMFA
0138	0138	APC610	PIF6	GETSTAT	BNZ
0139	0139	APC610	PIF6	GETSTAT	CMFA
0140	0140	APC610	PIF6	GETSTAT	BNZ
0141	0141	APC610	PIF6	GETSTAT	CMFA
0142	0142	APC610	PIF6	GETSTAT	BNZ
0143	0143	APC610	PIF6	GETSTAT	CMFA
0144	0144	APC610	PIF6	GETSTAT	BNZ
0145	0145	APC610	PIF6	GETSTAT	CMFA
0146	0146	APC610	PIF6	GETSTAT	BNZ
0147	0147	APC610	PIF6	GETSTAT	CMFA
0148	0148	APC610	PIF6	GETSTAT	BNZ
0149	0149	APC610	PIF6	GETSTAT	CMFA
0150	0150	APC610	PIF6	GETSTAT	BNZ
0151	0151	APC610	PIF6	GETSTAT	CMFA
0152	0152	APC610	PIF6	GETSTAT	BNZ
0153	0153	APC610	PIF6	GETSTAT	CMFA
0154	0154	APC610	PIF6	GETSTAT	BNZ
0155	0155	APC610	PIF6	GETSTAT	CMFA
0156	0156	APC610	PIF6	GETSTAT	BNZ
0157	0157	APC610	PIF6	GETSTAT	CMFA
0158	0158	APC610	PIF6	GETSTAT	BNZ
0159	0159	APC610	PIF6	GETSTAT	CMFA
0160	0160	APC610	PIF6	GETSTAT	BNZ
0161	0161	APC610	PIF6	GETSTAT	CMFA
0162	0162	APC610	PIF6	GETSTAT	BNZ
0163	0163	APC610	PIF6	GETSTAT	CMFA
0164	0164	APC610	PIF6	GETSTAT	BNZ
0165	0165	APC610	PIF6	GETSTAT	CMFA
0166	0166	APC610	PIF6	GETSTAT	BNZ
0167	0167	APC610	PIF6	GETSTAT	CMFA
0168	0168	APC610	PIF6	GETSTAT	BNZ
0169	0169	APC610	PIF6	GETSTAT	CMFA
0170	0170	APC610	PIF6	GETSTAT	BNZ
0171	0171	APC610	PIF6	GETSTAT	CMFA
0172	0172	APC610	PIF6	GETSTAT	BNZ
0173	0173	APC610	PIF6	GETSTAT	CMFA
0174	0174	APC610	PIF6	GETSTAT	BNZ
0175	0175	APC610	PIF6	GETSTAT	CMFA
0176	0176	APC610	PIF6	GETSTAT	BNZ
0177	0177	APC610	PIF6	GETSTAT	CMFA
0178	0178	APC610	PIF6	GETSTAT	BNZ
0179	0179	APC610	PIF6	GETSTAT	CMFA
0180	0180	APC610	PIF6	GETSTAT	BNZ
0181	0181	APC610	PIF6	GETSTAT	CMFA
0182	0182	APC610	PIF6	GETSTAT	BNZ
0183	0183	APC610	PIF6	GETSTAT	CMFA
0184	0184	APC610	PIF6	GETSTAT	BNZ
0185	0185	APC610	PIF6	GETSTAT	CMFA
0186	0186	APC610	PIF6	GETSTAT	BNZ
0187	0187	APC610	PIF6	GETSTAT	CMFA
0188	0188	APC610	PIF6	GETSTAT	BNZ
0189	0189	APC610	PIF6	GETSTAT	CMFA
0190	0190	APC610	PIF6	GETSTAT	BNZ
0191	0191	APC610	PIF6	GETSTAT	CMFA
0192	0192	APC610	PIF6	GETSTAT	BNZ
0193	0193	APC610	PIF6	GETSTAT	CMFA
0194	0194	APC610	PIF6	GETSTAT	BNZ
0195	0195	APC610	PIF6	GETSTAT	CMFA
0196	0196	APC610	PIF6	GETSTAT	BNZ
0197	0197	APC610	PIF6	GETSTAT	CMFA
0198	0198	APC610	PIF6	GETSTAT	BNZ
0199	0199	APC610	PIF6	GETSTAT	CMFA
0200	0200	APC610	PIF6	GETSTAT	BNZ
0201	0201	APC610	PIF6	GETSTAT	CMFA
0202	0202	APC610	PIF6	GETSTAT	BNZ
0203	0203	APC610	PIF6	GETSTAT	CMFA
0204	0204	APC610	PIF6	GETSTAT	BNZ
0205	0205	APC610	PIF6	GETSTAT	CMFA
0206	0206	APC610	PIF6	GETSTAT	BNZ
0207	0207	APC610	PIF6	GETSTAT	CMFA
0208	0208	APC610	PIF6	GETSTAT	BNZ
0209	0209	APC610	PIF6	GETSTAT	CMFA
0210	0210	APC610	PIF6	GETSTAT	BNZ
0211	0211	APC610	PIF6	GETSTAT	CMFA
0212	0212	APC610	PIF6	GETSTAT	BNZ
0213	0213	APC610	PIF6	GETSTAT	CMFA
0214	0214	APC610	PIF6	GETSTAT	BNZ
0215	0215	APC610	PIF6	GETSTAT	CMFA
0216	0216	APC610	PIF6	GETSTAT	BNZ
0217	0217	APC610	PIF6	GETSTAT	CMFA
0218	0218	APC610	PIF6	GETSTAT	BNZ
0219	0219	APC610	PIF6	GETSTAT	CMFA
0220	0220	APC610	PIF6	GETSTAT	BNZ
0221	0221	APC610	PIF6	GETSTAT	CMFA
0222	0222	APC610	PIF6	GETSTAT	BNZ
0223	0223	APC610	PIF6	GETSTAT	CMFA
0224	0224	APC610	PIF6	GETSTAT	BNZ
0225	0225	APC610	PIF6	GETSTAT	CMFA
0226	0226	APC610	PIF6	GETSTAT	BNZ
0227	0227	APC610	PIF6	GETSTAT	CMFA
0228	0228	APC610	PIF6	GETSTAT	BNZ
0229	0229	APC610	PIF6	GETSTAT	CMFA
0230	0230	APC610	PIF6	GETSTAT	BNZ
0231	0231	APC610	PIF6	GETSTAT	CMFA
0232	0232	APC610	PIF6	GETSTAT	BNZ
0233	0233	APC610	PIF6	GETSTAT	CMFA
0234	0234	APC610	PIF6	GETSTAT	BNZ
0235	0235	APC610	PIF6	GETSTAT	CMFA
0236	0236	APC610	PIF6	GETSTAT	BNZ
0237	0237	APC610	PIF6	GETSTAT	CMFA
0238	0238	APC610	PIF6	GETSTAT	BNZ
0239	0239	APC610	PIF6	GETSTAT	CMFA
0240	0240	APC610	PIF6	GETSTAT	BNZ
0241	0241	APC610	PIF6	GETSTAT	CMFA
0242	0242	APC610	PIF6	GETSTAT	BNZ
0243	0243	APC610	PIF6	GETSTAT	CMFA
0244	0244	APC610	PIF6	GETSTAT	BNZ
0245	0245	APC610	PIF6	GETSTAT	CMFA
0246	0246	APC610	PIF6	GETSTAT	BNZ
0247	0247	APC610	PIF6	GETSTAT	CMFA
0248	0248	APC610	PIF6	GETSTAT	BNZ
0249	0249	APC610	PIF6	GETSTAT	CMFA
0250	0250	APC610	PIF6	GETSTAT	BNZ
0251	0251	APC610	PIF6	GETSTAT	CMFA
0252	0252	APC610	PIF6	GETSTAT	BNZ
0253	0253	APC610	PIF6	GETSTAT	CMFA
0254	0254	APC610	PIF6	GETSTAT	BNZ
0255	0255	APC610	PIF6	GETSTAT	CMFA
0256	0256	APC610	PIF6	GETSTAT	BNZ
0257	0257	APC610	PIF6	GETSTAT	CMFA
0258	0258	APC610	PIF6	GETSTAT	BNZ
0259	0259	APC610	PIF6	GETSTAT	CMFA
0260	0260	APC610	PIF6	GETSTAT	BNZ
0261	0261	APC610	PIF6	GETSTAT	CMFA
0262	0262	APC610	PIF6	GETSTAT	BNZ
0263	0263	APC610	PIF6	GETSTAT	CMFA
0264	0264	APC610	PIF6	GETSTAT	BNZ
0265	0265	APC610	PIF6	GETSTAT	CMFA
0266	0266	APC610	PIF6	GETSTAT	BNZ
0267	0267	APC610	PIF6	GETSTAT	CMFA
0268	0268	APC610	PIF6	GETSTAT	BNZ
0269	0269	APC610	PIF6	GETSTAT	CMFA
0270	0270	APC610	PIF6	GETSTAT	BNZ
0271	0271	APC610	PIF6	GETSTAT	CMFA
0272	0272	APC610	PIF6	GETSTAT	BNZ
0273	0273	APC610	PIF6	GETSTAT	CMFA
0274	0274	APC610	PIF6	GETSTAT	BNZ
0275	0275	APC610	PIF6	GETSTAT	CMFA
0276	0276	APC610	PIF6	GETSTAT	BNZ
0277	0277	APC610	PIF6	GETSTAT	CMFA
0278	0278	APC610	PIF6	GETSTAT	BNZ
0279	0279	APC610	PIF6	GETSTAT	CMFA
0280	0280	APC610	PIF6	GETSTAT	BNZ
0281	0281	APC610	PIF6	GETSTAT	CMFA
0282	0282	APC610	PIF6	GETSTAT	BNZ
0283	0283	APC610	PIF6	GETSTAT	CMFA
0284	0284	APC610	PIF6	GETSTAT	BNZ
0285	0285	APC610	PIF6	GETSTAT	CMFA
0286	0286	APC610	PIF6	GETSTAT	BNZ
0287	0287	APC610	PIF6	GETSTAT	CMFA
0288	0288	APC610	PIF6	GETSTAT	BNZ
0289	0289	APC610	PIF6	GETSTAT	CMFA
0290	0290	APC610	PIF6	GETSTAT	BNZ
0291	0291	APC610	PIF6	GETSTAT	CMFA
0292	0292	APC610	PIF6	GETSTAT	BNZ
0293	0293	APC610	PIF6	GETSTAT	CMFA
0294	0294	APC610	PIF6	GETSTAT	BNZ
0295	0295	APC610	PIF6	GETSTAT	CMFA
0296	0296	APC610	PIF6	GETSTAT	BNZ
0297	0297	APC610	PIF6	GETSTAT	CMFA
0298	0298	APC610	PIF6	GETSTAT	BNZ
0299	0299	APC610	PIF6	GETSTAT	CMFA
0300	0300	APC610	PIF6		

COCO CADILLAC

Continued from Page 12

```
00205 0150 1000      BNA      GETS74      IS IT INVISIBLE ?
00206 015A 0101      CHPA      J1          ...NO
00207 015C 1002      BNE      GETS64
00208 015E 4F        CLRA
00209 0157 2006      BNA      GETS74
00210 0162 0103      CHPA      J3          SLOW BLINK ?
00211 0163 2002      BNE      GETS74      ...NO:DEFAULTS TO FAST BLINK
00212 0165 0001      LDA      J1
00213 0167 0E20      LDX      PD.AGS,V
00214 0169 0D01      STD      R0D,X
00215 016B 3F        CLRB
00216 016C 2004      BNA      GETEND
00217
00218 016E 0E20      GETS75      LDX      PD.AGS,V
00219 0170 0701      STA      R0A,X
00220 0172 3F        GETEND      CLRB
00221 0173 30          RTS          RETURN
00222
00223 0174 C0D0      GETS80      LDB      R0A,00000000 UNKNOWN SERVICE CALL
00224 0176 1A01      GETS80      ORCC      J1          SET CARRY ERROR!!!!!!
00225 0178 30          RTS
00226
00227 *****
00228 * PUTSTAT
00229 * FUNCTION KEY REPLACEMENT OF VALUES
00230 * IS THE ONLY ONE IMPLEMENTED
00231 *
00232 * A = CODE NO.
00233 * Y = ADDRESS OF PATH DESCRIPTOR
00234 * U = ADDRESS OF DEVICE STATIC STORAGE
00235 *
00236 * FOR FUNCKY
00237 *
00238 * B = NEW VALUE OF KEY
00239 * X = OFFSET OF KEY INTO TABLE
00240 *
00241
00242 0179 0196      PUTSTAT  CHPA      J150      IS IT A FUNCTION KEY CHANGE ?
00243 017B 1721      BEQ      PUTS40      FOR UNSHIFTED KEYS
00244
00245 017D 0197      CHPA      J151
00246 017F 2720      BEQ      PUTS54      FOR SHIFTED KEYS
00247 0181 0198      CHPA      J152
00248 0183 261F      BNE      GETS80      NO:BAD CALL...
00249 0185 11020249 PUTS80  LEAY      CTRFLG,PC GET CONTROL KEY TABLE
00250 0189 C137      PUTS10  CHPB      J55
00251 018B 22E7      BNE      GETS80      FAST TABLE END ?
00252 018D C130      CHPB      J0
00253 018F 25E3      BLO      GETS80      YES:SKIP(>55)
00254 0191 3404      PSWS      B          SAVE VALUE
00255 0193 1F10      TFR      X,D          PUT OFFSET INTO B
00256 0195 3502      PULS      A
00257 0197 1F21      TFR      Y,X          X POINTS TO TABLE
00258 0199 1A          ABX          AND IN KEY LOCATION
00259 019A 0704      STA      ,X          STORE NEW VALUE
00260 019C 5F        CLRB
00261 019D 30          RTS          NO ERROR
00262
00263 019E 110D0107 PUTS40  LEAY      KEYTAB,PC GET UNSHIFTED KEY TABLE
00264 01A2 2005      BRA      PUTS40      STORE VALUE
00265
00266 01A4 110D0109 PUTS50  LEAY      SHFTAB,PC GET SHIFTED KEY TABLE
00267 01A8 100F      BRA      PUTS40      STORE VALUE
00268 *****
00269 * TERMINATION ROUTINE
00270 *
00271 * SINCE THIS ROUTINE IS NEVER REALLY
00272 * PERFORMED IT IS NOT NECESSARY
00273 * BUT IS HERE JUST IN CASE
00274 *
00275
00276 01AA 1401      TERM      PSWS      CC          SET EVERYTHING TO NORMAL
00277 01AC 1A10      ORCC      J510
00278 01AE DC05      LDB      D.ALTRQ
00279 01B0 0D01      STD      D.IAQ
00280 01B2 3501      PULS      PC,CC
00281
00282 *****
00283 * READ ROUTINES
00284 *
00285 *
00286 * Y = PATH DESCRIPTOR
00287 * U=STATIC STORAGE
00288 * RETURNS CHAR IN A
00289 * OR ERROR IN B, CC SET ON ERROR
00290 *
00291
00292 01B4 30C034      READ      LEAX      INPBUF,U GET THE LOCATION OF THE INPUT
00293 01B7 06C027      LDB      BUFOUT,U GET LAST CHAR OUT
00294 01BA 1A50      ORCC      J550      DISABLE INTERPTS
00295 01BC 01C326      CNFB      BUFIN,U IF SAME AS LAST IN POSITION
00296 01BF 270F      BEQ      READ10      TELL CALLER AND WAIT
00297 01C1 3A          ABX          ELSE
00298 01C2 A604      LDA      ,X          GET THE NEXT CHARACTER
00299 01C4 5C          INCB
00300 01C5 C177      CNFB      #INPBUF-1
00301 01C7 2301      BLS      READ0
00302 01C9 5F        CLRB
00303 01CA 07C027      READ0     STB      BUFOUT,U SAVE OUTPUT POINTER
00304 01CD 1CAE      ANDCC      J5AE      RESTORE INTERRUPTS
00305 01C7 30          RTS
00306
00307 01D0 A644      READ10    LDA      V.BUSY,B TELL CALLER
00308 01D2 0740      STA      V.WAKE,U THE ROUTINE BUSY
00309 01D4 1C0F      ANDCC      J5AF
00310 01D6 0E0000      LDX      J50000      SLEEP FOREVER
00311 01D9 103F0A      ORS      J50000
00312 01DC 0F45      CLR      V.WAKE,U
00313 01DE 0E40      LDX      J50040      GET PROC DESCRIPTOR
00314 01E0 0E0026      LDB      J50,X
00315 01E2 370F      BEQ      READ      TRY TO READ AGAIN
00316 01E5 C103      CNFB      J502
00317 01E7 220E      BHL      READ      TRY READS FOR A WHILE
00318 01E9 43          COMA
00319 01EA 30          RTS
00320
00321 *****
00322 * KEYBOARD POLLING ROUTINES
00323 *
00324 *
00325
00326 01EB FE0040      POLKEY    LDB      >D.KBDSTA GET STATIC STORAGE
00327 01ED 0E001D      LDX      KPORT,U GET PORT ADDRESS
00328 01F1 0001      LDA      J,X          ANY DATA?
00329 01F3 2004      BNE      POL10      YES: LOOK FOR IT
```

```
00330 01F5 400F0020 JMB      J175 400F0020
00331 01F7 2A01      POL10     LDA      J,X
00332 01F9 00FF      LDA      J5FF
00333 01FB 0701      STA      J,X
00334 01FD 0701      LDA      J,X
00335 01FF 0004      COMA
00336 0201 43          ANDA      J501      REVERSE BITS
00337 0203 0403      BNE      JIGNOR      MASK OUT ALL BUT FIRES
00338 0205 2000      CLR      J,X          IGNORE IF FIRE BUTTON DOWN
00339 0207 0701      LDA      J,X          ELSE LOOK AGAIN
00340 0209 0004      COMA
00341 020A 43          ANDA      J5FF      REVERSE BITS
00342 020B 0477      BNE      GETKEY      MASK JOYSTICK ANALOG IN
00343 020D 2020      LDA      J5FF      GET THE KEY THATS DOWN
00344 020F 000F      LDA      J5FF      NO KEYS ARE DOWN
00345 0211 07C020      STA      KEYVAL,U CLEAR STORAGE
00346 0213 047D      LDA      J5FD      BUFFERS
00347 0215 0000      STA      LASTKY,U
00348 0217 0000      LDB      J50500      RESTORE REPEAT KEY VALUES
00349 0219 0000      STD      REPSPD,U
00350 021B 0000      LDA      >D.DSKTMR ADJUST FOR MOTOR TIMER
00351 021D 2700      BEQ      JIGEND      FOR CLOCK ADJUSTMENT
00352 021F 4A          DECA
00353 0221 0000      STA      >D.DSKTMR
00354 0223 2003      BNE      JIGEND
00355 0225 0000      STA      J5F40
00356 0227 0000      JMB      J175 400F0020
00357 0229 0000      BSR      FINDK      STROBE FOR KEY PRESSED
00358 022B 0000      CHPA      J5FF      CHECK FOR SHIFT LOCK
00359 022D 0000      BEQ      JIG00      IF NEGATIVE IGNORE...
00360 022F 0000      CHPA      J0
00361 0231 0000      BEQ      JIG01      IF ZERO IGNORE
00362 0233 0000      CHPA      J5FE      IS IT THE SHIFT LOCK ?
00363 0235 0000      BNE      STORE      NO THEN STORE THE CHAR
00364 0237 0000      COM      CAPLOC,U      YES:REVERSE LOCK CONDITION
00365 0239 0000      BRA      JIG01      RETURN
00366 023B 0000      LDB      BUFIN,U      CHECK INPUT BUFFER
00367 023D 0000      LEAX      INPBUF,U
00368 023F 0000      ABX
00369 0241 5C          INCB
00370 0243 0000      CNFB      #INPBUF-1
00371 0245 0000      BLS      STORE00      IF NOT FULL STORE THE KEY
00372 0247 0000      CLRB      STORE00      AND INCREMENT THE POSITION
00373 0249 0000      CHPA      J5FF      ELSE START BUFFER AT TOP
00374 024B 0000      CHPA      J5FF      SAVE NEW BUFFER POS
00375 024D 0000      BEQ      STORE10
00376 024F 0000      STB      BUFIN,U
00377 0251 0000      STA      ,X          SAVE CHAR
00378 0253 0000      BEQ      WAKE      WAKE CALLER IF CHAR=0
00379 0255 0000      CHPA      V.DCHR,U      PAUSE CHARACTER ?
00380 0257 0000      BNE      STORE00      NO:SKIP
00381 0259 0000      LDB      V.DEV2,U
00382 025B 0000      BEQ      WAKE      TELL CALLER TO PAUSE
00383 025D 0000      STA      V.PAUS,X
00384 025F 0000      BRA      WAKE
00385 0261 0000      LDB      #SSINTPT
00386 0263 0000      CHPA      V.INTPT,U KEYBOARD INTERRUPT SIGNAL
00387 0265 0000      BEQ      STORE00      INTERRUPT ?
00388 0267 0000      LDB      #SSABORT
00389 0269 0000      CHPA      V.QUIT,U YES:TELL LAST CALLER
00390 026B 0000      BNE      WAKE      ABORT SIGNAL
00391 026D 0000      LDA      V.LPRC,U CUIT ?
00392 026F 0000      BRA      WAKE      NO
00393 0271 0000      LDB      V.LPRC,U GET LAST PROCESS
00394 0273 0000      BRA      WAKE      AND WAKE IT
00395 0275 0000      LDB      J5WAKE      WAKE SIGNAL
00396 0277 0000      LDA      V.WAKE,U WAKE CALLER
00397 0279 0000      BEQ      WAKE00      NO CALLER TO WAKE
00398 027B 0000      ORS      J50000      CALLER TO WAKE
00399 027D 0000      CLR      V.WAKE,U RESET CALLER ID
00400 027F 0000      BRA      JIG20      FINISH
00401
00402 0280 4F          FINDK     CLRA
00403 0282 5F          CLRB          START THE LOCK CLEAN
00404 0284 0000      STA      COLCNT,U BY CLEARING OUT FLAGS ETC.
00405 0286 0000      STD      SHFLAG,U
00406 0288 0000      COMA
00407 028A 0000      BLS      FIN10
00408 028C 0000      STA      J,X          CLEAR LAST VALUE
00409 028E 0000      LDA      J,X          SEE IF THIS COLUMN IS ON
00410 0290 0000      COMA
00411 0292 047F      ANDA      J51F      MASK OUT JOY STICKS
00412 0294 0000      BEQ      NEXCOL      NOT ON:GET THE NEXT COLUMN
00413 0296 0000      LDB      J5FF      START FROM SCRATCH
00414 0298 0000      INCB      SHIFT ROW OVER LEFT
00415 029A 0000      LDB      J500
00416 029C 0000      BCC      FINDK0      IF KEY IS NOT FOUND
00417 029E 0000      BSR      MATCHR      IF KEY IS FOUND
00418 02A0 0000      CNFB      J506      MORE ROWS?
00419 02A2 0000      BCS      FINDK2      YES:LOOK MORE
00420 02A4 0000      INC      COLCNT,U ALL ROWS EXHAUSTED
00421 02A6 0000      ORCC      J501      SET CARRY
00422 02A8 0000      ROL      J,X          MOVE TO NEXT COLUMN
00423 02AA 0000      BCS      FINDK1      STILL NOT FOUND
00424 02AC 0000      LDA      KEY,U
00425 02AE 0000      BNE      NOKEY
00426 02B0 0000      CHPA      LASTKY,U
00427 02B2 0000      BEQ      REPEAT
00428 02B4 0000      STA      LASTKY,U
00429 02B6 0000      TFR      A,B
00430 02B8 0000      PSWS      X
00431 02BA 0000      LEAX      KEYTAB,PC
00432 02BC 0000      ABX
00433 02BE 0000      LDA      SHFLAG,U
00434 02C0 0000      BEQ      GETV10
00435 02C2 0000      LDB      J506
00436 02C4 0000      ABX
00437 02C6 0000      BRA      GETV20
00438 02C8 0000      LDA      CTRFLG,U
00439 02CA 0000      BEQ      GETV20
00440 02CC 0000      LDB      J112
00441 02CE 0000      ABX
00442 02D0 0000      LDA      ,X
00443 02D2 0000      CHPA      J505
00444 02D4 0000      BLO      GETV30
00445 02D6 0000      CNFB      J122
00446 02D8 0000      BHI      GETV30
00447 02DA 0000      CHPA      J91
00448 02DC 0000      BLO      UPPERC
00449 02DE 0000      CNFB      J96
00450 02E0 0000      BHI      LOWERC
00451 02E2 0000      STA      KEYVAL,U
00452 02E4 0000      PULS      X,PC
00453
00454 02F3 047D      NOKEY     LDA      J5FD
00455 02F5 07C020      STA      LASTKY,U
00456 02F7 0000      TST      SHFLAG,U
00457 02F9 0000      BNE      NOKEY0
00458 02FB 0000      TST      CTRFLG,U
00459 02FD 0000      BNE      NOKEY0
00460 02FF 0000      LDA      J5FF
00461 0301 30          RTS
00462 0303 30          RTS
00463
00464 0307 1404      MATCHR    PSWS      B          SAVE B
00465 0309 50          ASLB          MULTIPLY BY 8
00466 030B 50          ASLB
00467 030D 30          ASLB
```

Continued on Page 16

THE NEW COLOR COMPUTER

Continued from Page 1

by 24-line text display that is achieved by typing WIDTH 80. Add to that the eight background and foreground colors for the TEXT screen and a blinking cursor and underlining.

To best utilize the features of the Color Computer 3, Tandy has introduced a new RGB Analog monitor to take full advantage of the system hardware and provide 16 colors out of the palette of 64.

Other features include the documented 1.7 MHz clock; 32K ROM; 128K RAM (Standard, internally expandable to 512K); a 57-key keyboard including Control, Alternate, F1 and F2; text display capabilities of 32 x 16, 40 x 24 and 80 x 24; color graphics capabilities ranging from 64 x 32 (eight colors) to 640 x 192 (four colors) with six intermediate display formats including two displays that offer as much as 320 x 192 in 16 colors addressable through high resolution graphics. Extended BASIC and Program Paks.

I/O ports of the new machine include 1500 Baud cassette, two, 2-dimensional joysticks, RS-232C serial port, standard TV connector, composite video, RGB Analog and audio in addition to the standard ROM slot in the side.

As if this isn't enough, you have not heard the best news yet. This Color Computer lists for \$219.95! There's at least one vendor that is already listing this machine at the \$180 range.

When the Color Computer 3 is turned on, it has a few jobs to perform. It starts by jumping into the internal ROM at \$C000. This begins the process of copying the ROM to RAM and patching the ROM for the new commands. The new commands actually start at \$E000. Once completed, a short routine is copied into lower RAM where it then switches to the external ROM and checks for the presence of a DISK ROM and whether it is version 1.0 or 1.1 of RS-DOS.



OS-9 Users Group President Brian Lantz (left) and CompuServe SysOp Wayne Day discuss the new Tandy Color Computer at the press conference in New York.

Since the DOS must also be patched to access the new commands, this operation is both logical and practical. The problem comes when some other DOS is used, such as A-DOS or J-DOS. These enhanced operating systems tend to stretch past \$DFFF, which will overwrite the enhanced ECB commands. They also overwrite the check for DOS 1.0 and 1.1, so the Color Computer 3 is unable to determine the version of RS-DOS and defaults to none.

Some of the patches to ECB include the ability to LPEEK and LPOKE all 512K or RAM. This is the only area of BASIC that recognizes more than 128K, but may provide the information necessary for 512K spreadsheets in 80-columns.

Errors can now be trapped through the ON ERROR, ERNO and ERLIN commands. There's also an ON BREAK command.

The firebutton can now be recognized with the BUTTONN function,

removing the need to remember where to PEEK the firebutton.

The text screen is no longer limited to 32 x 16, which presents another problem. PRINT@ is not compatible in the 40 and 80-column modes. To fix this problem, the LOCATE(x,y) functions has been added to position the cursor on the screen. The remaining text screen commands are HSTAT (which gives the character, attributes and cursor information) and the ART command to set the text screen foreground and background colors, as well as the blinking and underline functions. All of the remaining Enhanced ECB commands are graphics oriented.

By this time next month, I hope to have had the time to run up my OS-9 Level II Color Computer and give you the run-down on what software runs and doesn't run. Until then . . . I think you'll love the newest addition to the Tandy line to home computers.

COMPUTERWARE[®] OS-9 Library

largest selection of OS-9 software anywhere!

Sourcerer

Menu driven symbolic EBOS disassembler produces symbolic source code that can be assembled with most editor-assemblers that use ASCII files.

CoCo \$39.95 SS-50 \$59.95

Advanced Utilities

KSMILL allows wildcards with any utility! Includes PATH for multiple path directories, UNLOAD for repetitive unlinks, FLINK for alternative boot files, ARCHIVE to backup large media to small media, & COPY.

CoCo \$29.95

Disk Fix & Utilities

Supports double-sided, double-density drives with 40 or 80 tracks & step rates to 6ms. Also includes DMDBE, DIRCOPY, PATCH, FILELOCK, & COMPARE. Supports both 1.X & 2.X.

CoCo \$34.95

Textools

Utilities for manipulating text files.

CoCo \$29.95 SS-50 \$49.95

Look & Listen

Hires screen is 27% faster & saves over 3.5K. Sound utilities for sound without hardware & driver, descriptor, & command for Radio Shack SSC. Not compatible with OS-9 2.0

CoCo \$29.95

Business Applications

Comprehensive General Ledger, Check Ledger, Inventory, Accounts Payable, Accounts Receivable, & Payroll.

To Order

Add \$2 surface or \$5 air shipping. Calif residents add 6% sales tax.

RAID Debugger

A screen oriented debugger with built-in disassembler, single step, memory window, snell access, & more!

CoCo \$39.95

Macro Conditional Assembler

Supports all standard mnemonics & directives plus thrives on macros, conditional assembly, repeat sequences, & source library files.

CoCo \$49.95 SS-50 \$59.95

Advanced Editor

Full featured line editor for programming or text. Edits files larger than memory.

CoCo \$39.95 SS-50 \$49.95

Text Formatter

Interfaces with any editor's ASCII text files to produce beautiful documentation.

CoCo \$34.95 SS-50 \$54.95

The Last Word

The ultimate screen word processor for OS-9. Fast on-screen formatting, pop-down menus, Word-Pak compatible. Edits files larger than memory.

CoCo \$49.95

OS-9 BASIC

Full featured, easy-to-use, high level language. Supports graphics & joystick for CoCo. By invoking OS-9 commands you can write utilities!

CoCo \$49.95 SS-50 \$74.95

Ask for our free catalog!



COMPUTERWARE
Box 668
Encinitas, CA 92024
(619) 436-3512



In New York, Falsoft President Lonnie Falk (left) and RAINBOW Managing Editor Jim Reed (right) look at the new Color Computer with Tandy Buyer Barry Thompson.

HEY, ALL YOU OTHER GUYS

by George Dorner

Let's get this straight at the outset. I like my CoCo! I really like the MOTD. I like the UG Software Library. In fact, I like almost everything about OS-9... especially the growth of the OS-9 user community. But there is a tiny concern I have which goes back to (scuse me while I stroke my beard) the days before Tandy embraced our favorite operating system and brought the broader exposure to OS-9 and the jump in membership for the UG.

Much of the appeal and much of the fun of the early (and tiny!) OS-9 community and Users Group were not due just to the knowledge that we were on to something good, something which was ahead of its time, something which was sophisticated and elegant. The breadth of the backgrounds of those involved and the scope of interests and applications of those using OS-9 intrigued most of us, I am sure. I have always been impressed at the great variety of folks I have met at the Microware seminars and even at the monthly meetings of our OS-9 group in Chicago. It's fascinating to rub elbows with users from major industrial firms, a major university, or with a bright individual with a good entrepreneurial idea which may use OS-9.

But I fret occasionally that, because most of the work being done for the UG just now is being done by users with at least one foot planted firmly in the CoCo world, some members or outsiders will get the feel that OS-9 is only for the CoCo. Worse, I worry that many OS-9 users may not experience the breadth and variety just alluded to and that this may limit or slow the growth of OS-9. That would be a shame, especially now that OS-9 is catching on in many quarters, and the number of applications is growing.

What to do to avoid "OS-9 myopia," the disease of only seeing one or two of the trees and missing the forest? Here's an antidote: Let's have more exposure for some of those OS-9 applications which most of us don't realize are going on. Let's hear about those VARs, OEMs, and OS-9 licensees who are outnumbered by the many CoCo users, SS-50 hobbyists, basement C hackers, who are so well represented here in the MOTD now. Let's hear the tales of how OS-9 is running on machines we didn't know existed, with software we would all give our left disk drive for, and in countries the names

of which we don't know how to pronounce. Tell us the stories of the research labs or corporate offices which use an OS-9 system innocuously and modestly outperforming a host of MS-DOS machines at some crucial tasks.

Well, one thing is sure. If nobody knows about such applications, nobody will write about them, and the myopia will continue. So, all you other guys, if you have an interesting or different OS-9 story to tell, dash off a few paragraphs and do yourself and the rest of us some good.

I know the stories are out there, and I will illustrate. I have talked to lots of OS-9ers in the past three years. I often end up talking long distance to some OS-9 zealot far away for long periods of time. One good member of the UG used to call me from various points around the country because he had no OS-9 contacts and had this lust to talk to someone . . . anyone! . . . about tree directories, pathnames, pipes, and whether OS-9 would ever have any graphics support. I still get those calls, and I regard that as a real bonus of my association with the OS-9 Home Group.

I've had a bunch of these contacts in the past week, and the breadth and variety of the OS-9 community really came through. One of these was a chat with a long-time acquaintance who is doing a development job on a system in which OS-9 will control 64 (count 'em!) 68000s. When this system is in production, you will probably be able to use one in your community . . . but without knowing that OS-9 is there. No more details until then.

I also talked to a developer who is putting together a 32-bit 68020 system on the VME buss with some really impressive performance benchmarks, 9-track tape support, and plans to market it as a business machine. These folks are hoping to beta test the 68020 version of OS-9 which has a new file manager for sequential blocks (SBFMAN?) and full memory management. Again, wait a couple of months for details.

A call came from Michigan from a CoCo hobbyist who had run his office on DynaStar and DynaCalc for over a year before installing a QT and moving over to Sculptor with great satisfaction both in the performance of the software and the hardware. His firm serves the auto industry, and while hearing a pitch from one of the biggies on a new

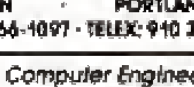
UNIX system he queried whether the UNIX system being quoted would outperform OS-9 in his application. The sales folks took a few moments to recall what OS-9 was, but then they acknowledged that OS-9 was probably faster for that application.

Finally, I just had the opportunity to use OS-9 on the IBM XT in our office! A friend called to ask if I would want to try out one of the TLM co-processor boards he had just purchased for their company. He didn't have to ask twice! We plugged it in and will report on our

findings next month. His firm is considering using it in a VAR application for real-time security monitoring, and they may place such a system with a customer so that a secretary can process words under MS-DOS while OS-9 is living in the same machine performing their applications. This is a real application of "If you can't beat 'em, . . ."

All this OS-9 news in just a week! OS-9 is indeed alive, and well, and growing. I have been a little close-mouthed in describing the projects since all of them are still under development. But I hope everyone will write up details when the smoke clears and the projects are in place. We all will be a little richer and a lot prouder for knowing more about OS-9 being used in scientific, industrial, business, or whatever applications. (I would like to hear more about the Australian music synthesizer using OS-9.)

05.9



LLOYD %
INC.

Lloyd % is a computer engineering corporation providing software and hardware products and consulting services.

19535 NE GUSAN PORTLAND, OR 97230 (USA)
PHONE: (503) 666-1097 TELEX: 910 380 5448 LLOYD %

Computer Engineers

K-BASIC™ IS HERE

K-BASIC is a TSC XBASIC (XPC) compatible COMILER
for OS/2 & FLEX price \$199

Here at last is a compiler for BASIC that will compile all your XBASIC programs. K-BASIC compiles TSC's KBASIC and KPC programs to machine code. K-BASIC is ready now to save you money and time by teaching your computer to

+ Think Faster + Conserve Memory + Be Friendlier

Call (503) 666-1097 for our CATALOG.

We have many programs for serious software development!

Micro BASIC for OS/2 \$149

A structured micro BASIC for general system control featuring Parameter passing, 10 string variables, 26 numeric variables, subroutines, nested loops, interactive I/O, sequential files, and time variables (for applications executing in the background) required to execute procedures such as disk or file backups. Includes the SEARCH and RESCUE UTILITIES™ (for OS/2 ONLY)

SEARCH and RESCUE UTILITIES™
 for OS/2 \$35

A super directory search utility. Output may be piped to the included utilities to perform file COPIES, DELETES, MOVES, LISTING (pagination), and FILTERING. Some filtering utility programs are included, of interest is the FILE DATE CHECKING utilities YOUNGER and DRAFT (Level 2) (for OS/2 Level 1 and 2)

PATCH™
 Modern Communications for OS/2 \$39

PATCH is a modern communications program for OS/2 featuring KEY MACROS, ASCII TEXT AND BINARY FILE UP/DOWN LOADING, PRINTER COPY, and HELP MENUS. We use it several times each day with our TELEX service. PATCH is convenient and easy to use. Key macros may be pre-stored and loaded at any time.

CRASMB™
CROSS ASSEMBLER PACKAGE
 for OS/2 & FLEX... all for \$399
 Motorola CPUs... \$150
 Intel CPUs... \$150, Others... \$150

CRASMB is the highly acclaimed cross assembler package for OS/2 and FLEX systems. It runs on our OS/2 development station or development station. It is available on

NOW FOR OS/2 \$432

CRASMB features: 16-bit Constant Long symbol names
 Symbol cross reference table
 Code in 4 formats (OS/2, FLEX, S-1-S*, INTEL HEX)

USA: England: Germany: Australia: Japan: Switzerland: Sweden:	VISA MC, CDD, CHECKS, ACCEPTED (LLOYD %) (503) 666-1097 VISAway (503) 473-4951 Sachai Computer (503) 255-2991 Park Radio Electronics (344) 9111 Microboards (9474) 22-1741 Etron AG (089) 36 27 24 Microstar Scandinavien AG (018) 123595	S.T. MEDIA (800) 338-6600 Windus (0692) 405189 K&H Software (06983) 6748 Sekko (03) 832-4000
---	--	---

* Radio, Inc. S.A. (LLOYD %) (503) 666-1097
 PATCH CRASMB and CRASMB V. 1.0 are trademarks of LLOYD %
 OS/2 is a trademark of International Business Machines Corporation

Continued from Page 13

[illegible]

Continued on Page 17

COCO CADILLAC

Continued from Page 16

```
##732  #544 8DA8      BSR  UPDWP      IN CRTC
##733  #544 39
##734
##735  #547 2E0000    REMOVL  LDY  #0      HOME AND CLEAR THE SCREEN
##736  #54A A7C82F    STX  DSTART,U  ZERO SCREEN START
##737  #54D 10E00000  LDY  #00000    LENGTH OF SCREEN
##738  #551 17FF7E    LBSR  CLEAR
##739  #554 4F        CLR    CLR    SET POSITION TO
##740  #555 A7C82D    STA  CURLIN,D  TOP LINE
##741  #558 A7C82E    STA  CURCOL,U  COLUMN #
##742  #55B A7C833    LDY  DSTART,U  SET CRTC
##743  #55E C6FC      LDB  FPC      NEW TOP
##744  #561 8D8C      BSR  DDOWN
##745  #562 C417      LDB  #512     SET POSITION
##746  #564 4D60      BSR  UPDWP
##747  #566 C6FC      LDB  #5FE     AND CURSOR
##748  #568 4D84      BSR  UPDWP
##749  #56A 3F        CLR    CLR    NO ERRORS
##750  #56B 39      RTS
##751
##752  #56C 81E3      ERLINE  CMPA  #1  IS IT ERASE WHOLE LINE?
##753  #56E 2605      BNE  ERLINE    NO:
##754  #570 4FC022    CLR  CURCOL,U  YES: START AT THE BEGINNING
##755  #573 C854      LDB  #0        CLEAR TO END
##756  #575 E00022    STDB  CURCOL,U  COUNT IS NOW IN B
##757  #578 4F        CLR    CLR
##758  #579 1F02      TFR  D,Y      PUT COUNT INTO Y
##759  #57B 8D24      BSR  FINDPO   GET THE POSITION
##760  #57D 16F752    LBR    CLEAR  CLEAR THE SCREEN
##761
##762  #580 8D17      BSR  FINDPO   GET POSITION
##763  #582 A7C835    STX  WORK,U   SAVE IT
##764  #585 E0C033    LDB  DSTART,U  FIND SCREEN END
##765  #588 C30700    ADDD  #00000
##766  #58B A3C322    SUBD  WORK,D  ADJUST CLEAR COUNT
##767  #58E 1F02      TFR  D,Y      PUT COUNT IN Y
##768  #590 16F737    LBR    CLEAR  CLEAR THE SCREEN
##769
##770  #591 8D8C      BSR  FINDPO   MOVE THE CURSOR
##771  #595 C00E      LDB  #000     TO THE CURRENT POSITION
##772  #597 17F754    LBSR  UPDWP
##773  #59A C018      LDB  #112     SET POSITION
##774  #59C 17F74F    LBSR  UPDWP
##775  #59F 5F        CLR    CLR
##776  #5A0 39      RTS
##777
##778  #5A1 A6C82D    FINDPO  LDA  CURLIN,U  GET LINE #
##779  #5A4 C050      LDB  #0       AND FIGURE THE POS
##780  #5A6 30        HUC    HUC    IN SCREEN
##781  #5A7 E1C033    ADDD  DSTART,D  MEMORY
##782  #5AA 1F01      TFR  D,X      PUT IT IN X
##783  #5AC E6C82E    LDB  ADD IN THE COL
##784  #5AF 3A        ABX    ABX    COUNT TOO
##785  #5B0 30        RTS    RETURN
##786
##787  #5B1 A6C82E    CURLFT  LDA  CURCOL,D  BACKSPACE
##788  #5B4 40        TSTA  ADO START OF LINE?
##789  #5B5 260C      BNE  CURLFT   NO
##790  #5B7 8E50      LDB  #0       YES: MOVE TO END
##791  #5B9 E6C82D    LDB  CURLIN,U  LEFT ONE POSITION
##792  #5BC 5D        TSTB  ADO 7
##793  #5BD 270A      BEQ  CURLFT   YES/SKIP LINE CHANGE
##794  #5B7 5A        DECB  ADO     GO UP ONE LINE
##795  #5C0 E7C82D    STB  CURLIN,D  SAVE IT
##796  #5C3 4A        DECB  ADO     GO BACK ONE CHARACTER
##797  #5C4 A7C82E    STA  CURCOL,U  SAVE NEW COLUMN
##798  #5C7 20CA      BRA  NOVCUR   MOVE THE CURSOR
##799  #5C9 51        CLR    CLR
##800  #5CA 39      RTS
##801
##802  #5CB 60C83D    CURUP   TST  CURLIN,U  MOVE UP ONE LINE
##803  #5CE 27F9      BEQ  CURUP    ALREADY AT TOP SKIP
##804  #5D0 6AC82D    DEC  CURLIN,U  MOVE UP
##805  #5D3 20BE      BRA  NOVCUR
##806
##807  #5D5 E6C83D    CURDOWN LDB  CURLIN,U  LINEFEED
##808  #5D8 5C        INCB  INCB
##809  #5DA C118      CMPB  CMPB  #24
##810  #5DB 2B05      BMI  CURDIN  FAST SCREEN END ?
##811  #5DD 17F743    LBSR  ADO     NO:IGNORE
##812  #5DE 2003      BNA  ADO     ELSE SCROLL
##813  #5E0 2003      BNA  CURDIN  LINE # SAME
##814  #5E2 E7C82D    CURDIN  STB  CURLIN,U  SAVE NEW LINE #
##815  #5E5 20AC      BRA  NOVCUR   MOVE THE CURSOR
##816
##816  #5E7 60C83A    ESCAPE  TST  ESCCNT,U  TEST COUNT
##817  #5EA 2658      BNE  ESCAPE  GO IF NOT 0
##818  #5EC 4141      CMPA  #000    IS IT A CLEAR TO END
##819  #5EE 2605      BNE  ESCAPE  NO:SKIP
##820  #5F0 8604      LDA  #0       YES TRANSATE AND EXECUTE
##821  #5F2 16FF77    LBR    ERLINE
##822  #5F5 8143      CMPA  #543    IS IT A CLEAR TO END
##823  #5F7 2787      BEQ  EXTEND  YES:GO DO IT
##824  #5F9 3156      CMPA  #556    IS IT A BLACK CURSOR
##825  #5FB 2610      BNE  ESCAPE  NO:SKIP
##826  #5FD A60DFA35  LDA  CURLIST,PCR  CHANGE CURSOR
##827  #601 84F8      ANDA  #11111111
##828  #603 1F01      TFR  D,X      UPDATE CURSOR
##829  #605 C60A      LDB  #1F     TYPE REG
##830  #607 17FEE4    LBSR  UPDWP
##831  #60A 160109    LBR    CTREND  CLEAR ESCAPE FLAG
##832  #60D 8176      CMPA  #076    UNDERLINE CURSOR ?
##833  #60F 2600      BNE  ESCAPE  NO:SKIP
##834  #611 E000FAA1    LDB  CURLIST,PCR  CHANGE CURSOR
##835  #615 8A07      ORA  #100000111
##836  #617 20E2      BRA  ESCAPE
##837  #619 0153      CMPA  #053    SET INVERS/NORMAL?
##838  #61B 2600      BNE  ESCAPE  NO:SKIP
##839  #61D 60C83A    ESCAPE  INC  ESCCNT,U  YES WAIT FOR SECOND CHAR
##840  #620 A7C82C    STA  ESCBUF,U  STORE CODE FOR NEXT PROCESS
##841  #623 5F        CLR    CLR
##842  #624 39      RTS    RETURN NO ERRORS
##843  #625 8157      ESCAPE  CMPA  #557    SET CURSOR RATE?
##844  #627 27F4      BEQ  ESCAPE  YES/WAIT FOR NEXT CHAR
##845  #629 8149      CMPA  #049    INSERT A LINE
##846  #62B 102710C   LBSR  DYNLIN  GO DO IT
##847  #62F 8146      CMPA  #546    DELETE A LINE
##848  #631 102710C   LBSR  DYNLIN  GO DO IT
##849  #635 8147      CMPA  #547    SHIFT LINE RIGHT 8
##850  #637 102701AE  LBSR  SHFLIN  YES
##851  #63B 8148      CMPA  #548    SHIFT LINE LEFT 8
##852  #63D 102701A8  LBSR  SHFLIN  YES
##853  #641 1600D2    LBR    CTREND  ELSE POOR TRY
##854
##855  #644 E6C81C    DOESC   LDB  ESCBUF,U  FIND OUT WHAT'S WAITING
##856  #647 C153      CMPB  #53     IS IT A REVERSE VIDEO ?
##857  #649 2610      BNE  DOESC    NO:SKIP
##858  #64B 8120      CMPA  #020    NORMAL MODE ?
```

```
##859  #64D 2607      BNE  DOESC    NO SKIP
##860  #64F 47        CLR    CLR    ELSE SET FLAG FOR NORMAL
##861  #650 A7C827    STA  INVFLG,U  AND SAVE IT
##862  #653 1600D9    LBR    CTREND  CLEAR ESCAPE
##863  #656 8121      CMPA  #021    IS IT INVERSE?
##864  #658 102600A   LBR    CTREND  NO:BAD CALL-CLEAR IT
##865  #65C 86FF      LDA  #11111111  ELSE SET FOR INVERSE
##866  #65E A7C82F    STA  INVFLG,U  SAVE IN FLAG
##867  #661 1600D9    LBR    CTREND  CLEAR ESCAPE
##868  #664 C157      CMPB  #057    IS IT A CHANGE IN CURSOR RAT
##869  #666 102600AC  LBR    CTREND  NO:CLEAR ESCAPE
##870  #66A 8120      CMPA  #020    STEADY CURSOR?
##871  #66C 2600      BNE  DOESC    NO:TRY NEXT
##872  #66E A60DFA44  LDA  CURLIST,PCR  CHANGE CURSOR
##873  #672 840F      ANDA  #100001111
##874  #674 1600D2    LBR    CURSIF  AND STORE IT
##875  #677 8122      CMPA  #022    FAST BLINK?
##876  #679 2600      BNE  DOESC    NO:...
##877  #67B E000FA37  LDB  CURLIST,PCR  MAKE A FAST BLINK
##878  #67F 840F      ANDA  #100001111
##879  #681 8000      ADDA  #000
##880  #683 1600D3    LBR    CURSIF
##881  #686 8122      CMPA  #022    INVISIBLE ?
##882  #688 260A      BNE  DOESC    NO
##883  #68A E000FA28  LDA  CURLIST,PCR  MAKE INVISIBLE
##884  #68E 840F      ANDA  #100001111
##885  #690 1625      ADDA  #020
##886  #692 2075      BRA  CURSIF
##887  #694 E000FA1E  DOESC   LDB  CURLIST,PCR  DEFAULT IS A SLOW BLINK
##888  #696 840F      ANDA  #100001111
##889  #698 2600      ADDA  #000
##890  #69A 2600      BRA  CURSIF
##891
##892  #69E 84C82A    FUNKEY  LDB  ESCCNT,U  ALL CHARACTERS READ
##893  #6A1 C101      CMPB  #01     NO:
##894  #6A3 1700      BEQ  FUNKEY   YES: CHANGE KEY
##895  #6A5 5C        INCB  INCB    NO: INCR COUNT
##896  #6A6 E7C82A    STB  ESCCNT,U  AND STORE IT
##897  #6A9 A7C82C    STA  ESCBUF,U  AND CODE
##898  #6AC 3F        CLR    CLR    RETURN NO ERROR
##899  #6AD 39      RTS
##900  #6AE 1700      FUNKEY  TFR  A,B      YES PUT OFFSET IN B
##901  #6B0 47        CLR    CLR
##902  #6B1 1701      TFR  D,X
##903  #6B3 A6C82B    LDA  ESCFLG,D  PUT NEW CODE INTO A
##904  #6B6 811D      CMPA  #51D    UNSHIFTED KEY CHANGE ?
##905  #6B8 270C      BEQ  FUNKEY   ...YES
##906  #6BA 811E      CMPA  #51E    SHIFTED KEY ?
##907  #6BD 2710      BEQ  FUNKEY   ...YES
##908  #6BE E6C82C    LDB  ESCBUF,D
##909  #6C1 17FAC1    LBSR  PUTS40   AND CHANGE KEY
##910  #6C4 2050      BRA  CTREND  CLEAR CONTROL CALL
##911
##912  #6CE 26C82C    FUNKEY  LDB  ESCBUF,U
##913  #6D0 17FAD2    LBSR  PUTS40   SAVE UNSHIFTED VALUE
##914  #6D3 3F48      BRA  CTREND
##915
##916  #6DE E6C83C    FUNKEY  LDB  ESCBUF,U
##917  #6E1 17FAD0    LBSR  PUTS50   SAVE SHIFTED VALUE
##918  #6E4 2040      BRA  CTREND
##919
##920  #6E6 E6C82A    XYPOS   LDB  ESCCNT,U  GET COUNT
##921  #6E9 C101      CMPB  #01     ALL VALUES GATHERED ?
##922  #6EB 1700      BEQ  XYPOS    YES:MOVE THE CURSOR
##923  #6ED A7C82C    STA  ESCBUF,U  NO: SAVE VALUE
##924  #6EF 5C        INCB  INCB    INCREMENT COUNT
##925  #6F1 E7C82A    STB  ESCCNT,U  AND STORE
##926  #6F4 5F        CLR    CLR    RETURN NO ERRORS
##927  #6F7 39      RTS
##928
##928  #6E6 8020      XYPOS   SUBA  #020  ADJUST VALUE
##929  #6E9 A7C83D    STA  CURLIN,U  STORE Y
##930  #6EB A6C82C    LDA  ESCBUF,U  GET X
##931  #6EE 8020      SUBA  #020  ADJUST
##932  #6F1 A7C82E    STA  CURCOL,U  STORE
##933  #6F4 17F00D    LBSR  NOVCUR  MOVE THE CURSOR
##934  #6F6 201E      BRA  CTREND
##935
*****
* INSTEAD OF COLORS FOR #5 RM
* THIS CHANGES SHAPES AND BLINKS
* ONLY ALTERNATIVE THAT DOES SOMETHING
*****
##941  #6F8 8020      CURSOR  SUBA  #020  STRIP OFFSET
##942  #6F1 201A      BNE  CTREND  IF NOT IN RANGE
##943  #6FC 8100      CMPA  #00     NO:
##944  #6FE 2216      BNE  CTREND  IGNORE COMMAND
##945  #700 1F00      TFR  A,B      PUT INTO B
##946  #702 00000010  LEAX  CURSEL,PCR  TO FIND TYPE
##947  #705 3A        ABI  ABI
##948  #707 A60A      LDA  #X      IN TABLE
##949  #709 C000      LDB  #0      GET CURSOR NES
##950  #70B 1F01      TFR  D,X      GET NORMAL, LSB
##951  #70D A7C82C    STX  CURLIST,PCR  PUT IT IN X
##952  #711 C00A      LDB  #0A     SAVE NEW CURSOR
##953  #713 17F00D    LBSR  UPDWP   UPDATE CURSOR REG
##954  #716 5F        CLR    CLR    IN CRTC
##955  #717 E7C82B    CTREND  STA  ESCFLG,U  CLEAR ALL ESCAPE FLAGS
##956  #71A E7C82A    STB  ESCCNT,U  AND COUNTERS
##957  #71D 39      RTS    AND RETURN
##958
##958  #71E 606100    CURSEL  PCB  #00,$07,B  SLOW BLOCK,SLOW UL,STEADY BL
##959  #721 404077      PCB  #00,$07,B  FAST BLK, FAST UL,STAY UL
##960  #724 6444      PCB  #00,$04  SLOW HALF BLK,FAST 1/2 BLK
##961  #728 6546      PCB  #00,$04  SLOW 3/4 BLK,FAST 3/4 BLOCK
##962
*****
* JOYSTICK STATUS
*
* X - B FOR RIGHT STICK
* 1 FOR LEFT
*
* RETURNS :
* X = X VALUE
* Y = Y VALUE
* A = A FOR FIRE OFF,$7F FOR ON
*****
##974  #72A AE2E      JOYSTK  LDY  #0,REG,Y  CHECK CALLER'S REGS
##975  #72C 1021      PSNB  #1,CC  SAVE CC AND DESCRIPTOR ADDR
##976  #72E 1A10      ORCC  #010    DISABLE INTERRUPTS
##977  #730 80FF      LDA  #FFF      CHECK FOR FIRE
##978  #732 7FFF02    LBR  #733 0FF700
##979  #734 10210004  LDY  #0,X,X      LEFT STICK
##980  #736 2605      BNE  JOYSIF   YES:GO
##981  #738 C001      ANDB  #001     SET FOR RIGHT FIRE
##982  #73A C00200    BNE  JOYSIF   ON
##983  #73C 2005      BRA  JOYSIF   NOT ON
##984
##984  #740 47        JOYSIF  CLR    CLR    NO FIRE
##985  #742 A701      STA  #0A,X      STORE FIRE
##986  #744 84FF03    LDA  #7FF03    GET X
##987  #746 8A00      DRA  #000      POSITION
##988  #748 10A004    LDY  #0A,X      GET OLD X VALUE
##989  #74A 2602      BNE  JOYSIF   IF RIGHT
##990  #74C 84F7      ANDA  #5F7     ELSE MASK
```

Continued on Page 18

Continued from Page 17

```

01057 0700 1F00 IFA B,X
01058 070F 5F CLRS NO ERRORS
01059 0700 39 RTS
01060
01061 0701 100E0110 BILL LDY 0000 RING THE BELL; TIME COUNT
01062 0705 00FF22 BILL1# LDA 0FF22 USING SINGLE BIT SOUND
01063 0708 0A00 ORA 00000001# SET UP FOR HIGH PART
01064 070A 07FF22 STA 0FF22 OF SQUARE WAVE
01065 070D 000010 LDY 00# DURATION FOR PITCH
01066 0700 001F BILL2# LEAX -1,X COUNT IT DOWN
01067 0702 00FC BNE BILL2#
01068 0704 00FF22 LDA 0FF22 THEN SOUND LOW PART
01069 0707 04FD ANDR 00111111#1 OF WAVE
01070 0709 07FF22 STA 0FF22
01071 070C 000010 LDY 00# COUNT AGAIN
01072 0707 001F BILL3# LEAX -1,X UNTILL DONE
01073 07E1 00FC BNE BILL3#
01074 07E1 013F LEAY -1,Y DONE ENOUGH?
01075 07E5 00DE BNE BILL3# NO;RING SOME MORE
01076 07E7 0F CLRS NO ERRORS
01077 07EB 39 RTS
01078
01079 *****
01080 * ROUTINES FOR CP/M COMPATIBILITY
01081 *
01082 *
01083 *****
01084 07E9 3402 BRFLIN PSRS A SHIFT LINE 5 POSITIONS
01085 07EB 0023 BSR LINSTN GET START OF LINE
01086 07ED C000 LDR #0 HOW MUCH TO MOVE
01087 07EF 000040 LEAX 72,X ENO
01088 07F1 3502 PULS A GET BACK CODE
01089 07F4 0147 CMPA 0547 MOVE RIGHT?
01090 07F6 7724 BEQ MOVHGT YES
01091 07F8 2004 BRA MOVHFT NO; MOVE IT LEFT
01092
01093 07FA 3402 DINTLN PSRS A DELETE OR INSERT A LINE
01094 07FC 0012 BSR LINSTN GET CURRENT LINE START
01095 07FE 200030 LDR 0START,V GET LAST LINE
01096 0801 C00730 ADDD 0573#
01097 0804 1F01 TFR D,X STORE IT
01098 0808 C050 LDR 00# AMOUNT TO MOVE
01099 080B 0502 PULS A
01100 080A 0145 CMPA 0545 INSERT A LINE
01101 080C 270E BEQ MOVHGT YES
01102 080E 2012 BRA MOVHFT ELSE DELETE A LINE
01103
01104 0810 17FD0E LINSTN LBSR FINDPO GET CURRENT POSITION
01105 0812 E6C03E LDR CURCOL,D DELETE COLUMNS
01106 0826 50 HEBE
01107 0817 3005 LEAX B,X TO FIND LINE START
01108 0819 1712 TFR X,Y PUT IT IN Y
01109 082B 39 RTS

```

```

#1110
#1111      * SHIFT LEFT
#1112      * B = NO. OF CHARS
#1113      * Y = START POS
#1114      * X = END POS
#1115
#1116
#1117 #B1C 18AFCB32  MOVWGT  STX  WORK,U      SAVE START
#1118 #B2B 3B1F      LEAX  -1,X      NEEDED FOR ALIGNMENT
#1119 #B22 34B4      MOVW1B  PSWS  B      SAVE COUNT
#1120 #B24 17FD73    LEBS  MOVCI1B  MOVE CURSOR TO POS
#1121 #B27 17FC9D    LEBS  GETCHA  GET A CHARACTER
#1122 #B2A 35B4      PULS  B      GET OFFSET
#1123 #B2C 3414      PSWS  B,X      AND TO POS
#1124 #B2E 3A      ABRX
#1125 #B2F 34B2      PSWS  A      SAVE CHAR
#1126 #B31 17FD66    LEBS  MOVCI1B  MOVE TO POS
#1127 #B34 35B7      PULS  A      GET BACK AND
#1128 #B36 17FC6D    LEBS  FUTCHA  MOVE TO NEW LOCATION
#1129 #B39 3514      PULS  B,X      GET BACK OLD POS
#1130 #B3B 1B1F      LEAX  -1,X      DECR POS
#1131 #B3D ACCB32    CMPLX  WORK,U      DONE ?
#1132 #B4B 2CEB      BGE  MOVW1B  NO
#1133 #B42 32BFBF    MOVW9B  LBRX  #B      CLEAR POSITION
#1134 #B45 3A      ABRX
#1135 #B46 1E12      EXG  X,Y      IS Y
#1136 #B4B 17FC67    LEBS  CLEAR  GO
#1137 #B4B 16FCEB    LEBS  CTEND   ALL DONE;CLEAR CONTROLS
#1138
#1139      *****
#1140      * SHIFT LEFT
#1141      * B = NO. OF CHARS
#1142      * Y = START POS
#1143      * X = END POS
#1144
#1145
#1146 #B4E A7CB32  MOVWLT  STX  WORK,U      THIS DOES JUST WHAT
#1147 #B51 1E12    EXG  X,Y      MOVWGT, ONLY IN REVERSE
#1148 #B53 3414  MOVW1B  PSWS  B,X
#1149 #B56 3A      ABRX
#1150 #B56 17FD41    LEBS  MOVCI1B
#1151 #B59 17FC6B    LEBS  GETCHA  GET A CHARACTER
#1152 #B5C 3514      PULS  B,X
#1153 #B5E 34B4      PSWS  B
#1154 #B60 34B2      PSWS  A
#1155 #B62 17FD15    LEBS  MOVCI1B
#1156 #B65 35B2      PULS  A
#1157 #B67 17FC1C    LEBS  FUTCHA  SAVE B CHARACTER BACK
#1158 #B6A 35B4      PULS  B
#1159 #B6C 3B81      LEAX  1,X
#1160 #B6E ACCB32    CMPLX  WORK,U
#1161 #B71 25E0      BLO  MOVW1B
#1162 #B73 28CB      BRX  MOVW9B
#1163
#1164 #B75 3ED266    MOVW1B  EXOD
#1165 #B78          WDR0F1  EQU  *
#1166
#1167

```

```

00000 error(s)
00015 warning(s)
00078 0160 program bytes generated
00097 00151 data bytes allocated
02010 10522 bytes used for symbols

```